UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Telecomunicació
i Aeroespacial de Castelldefels

TREBALL DE FI DE GRAU

TFG TITLE: CFD study of the influence of mountain and valley shapes on wind propagation

DEGREE: Grau en Enginyeria d'Aeronavegació

AUTHOR: Pedro López Gallego

ADVISORS: Santiago Arias Calderón
Adeline De Villardi De Montlaur

DATE: July 4, 2019

Título: CFD study of the influence of mountain and valley shapes on wind propagation
Autor: Pedro López Gallego

Directores: Santiago Arias Calderón
Adeline De Villardi De Montlaur

Fecha: 4 de juliol de 2019

Resumen

En el siguiente trabajo se explica un estudio aerodinámico, que consiste en ver la influencia que tiene la superficie terrestre sobre la propagación del viento. Concretamente en llanuras, pendientes de subida y de bajada, geometrías básicas que representan desniveles terrenales de zonas montañosas. Una vez el viento atraviesa los dominios mencionados, se analiza el perfil de velocidad resultante. Todo con tal de proporcionar evidencia científica a las investigaciones de la energía eólica, mostrando las alturas en donde un aerogenerador aprovecha la máxima energía cinética del viento. Para hacerlo posible, se utiliza un programa libre y comercial en Dinámica de Fluidos Computacional (CFD) denominado OpenFOAM. En él, se especifican las condiciones físicas de los contornos de cada geometría. Como la entrada del flujo, el terreno por donde se propaga, su salida y la limitación superior. Se examinan las condiciones de contorno y se concluyen cuáles son las más adecuadas para la tesis. La rugosidad se simula mediante las "funciones de pared", las cuales representan la obstaculización del terreno. La turbulencia que aparece en consecuencia, se analiza mediante el modelo RANS, concretamente el de k-épsilon. Y con tal de reducir factores que puedan aumentar dicha turbulencia, la velocidad de entrada que se impone es reducida.

Conocimientos físicos sobre la Capa Límite Atmosférica se tienen en cuenta en el trabajo. Además de tener como referencia los estudios realizados por Richard y Hoxey en el 1993, los cuales modelaron un perfil de velocidad teórico adaptado a las condiciones de la Capa Límite Atmosférica. Modelo que se usa como velocidad teórica de cada simulación, por lo que los resultados se compararán con éste.

Finalmente se discuten los resultados obtenidos, así como otros factores que se tienen en cuenta a lo largo del trabajo. El estudio termina con unas conclusiones sobre los resultados y el objetivo de la investigación planteada.

Title : CFD study of the influence of mountain and valley shapes on wind propagation
Author: Pedro López Gallego

Advisors: Santiago Arias Calderón
Adeline De Villardi De Montlaur

Date: July 4, 2019

Overview

In the following work an aerodynamic study is explained, which consists in seeing the influence of the earth's surface on the wind propagation. Specifically in plains, uphill and downhill slopes, basic geometries that represent terrestrial and mountainous areas. Once the wind passes through the mentioned domains, the resulting velocity profile is analysed. All in order to provide scientific evidence to the investigations of wind energy, showing the heights where a wind turbine takes advantage of the maximum kinetic energy of the wind. To make it possible, a free and commercial program in Computational Fluid Dynamics (CFD) called OpenFOAM is used. In it, the physical conditions of the boundaries of each geometry are specified. Like the inlet of the flow, the land through which it spreads, its outlet and the top limitation. The boundary conditions are examined and the most suitable ones for this work are concluded. The roughness is simulated by the "wall functions", which represent the obstacle of the terrain. The turbulence that appears in consequence, is analysed by the RANS model, specifically that of k-epsilon. And in order to reduce factors that may increase this turbulence, the inlet velocity that is imposed is reduced.

Physical knowledge about the Atmospheric Boundary Layer is taken into account in the work. In addition to having as reference the studies carried out by Richard and Hoxey in 1993, which modelled a theoretical velocity profile adapted to the conditions of the Atmospheric Boundary Layer. Model that is used as the theoretical velocity of each simulation, so the results will be compared with it.

Finally, the results obtained are discussed, as well as other factors that are taken into account throughout the work. The study ends with some conclusions about the results and the objective of the proposed research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The use of renewable energies has had a great impact on the scientific world. Humans have been able to transform the energy that surrounds us into a useful energy in the engineering sector. The movement of air is a phenomenon produced by nature and humans are capable of transforming their kinetic energy into electricity. In this way, it can be taken advantage of an energy generated freely, in an energy that can be used for the needs of living beings. And in the case of wind, this is achieved through the use of windmills or wind turbines.

The strategy of the wind turbines, consists of placing a series of blades together on a rotating shaft, which turns and generates electricity when they are hit by the force of the wind. So, at more speed, more amount of movement and therefore more electricity will produce the wind turbine. Therefore, knowing the behaviour of the wind over any obstacle is very important to know how to properly position a wind turbine. Knowledge about fluid mechanics and aerodynamics are necessary for this type of research.

Objective

The objective of this work is to study the influence of mountains and valleys shapes on the propagation of wind. Study which could provide scientific evidence to research on the improvement of wind energy. All this through a commercial program in CFD OpenFOAM, which will be able to simulate the roughness of the land proposed and it will simulate the development of the wind through time.

Overview

- Chapter 1: Basic Wind Principles. Basic concepts of wind are specified. As well as those of the Atmospheric Boundary Layer. The objective is to explain the velocity profile that will be modelled in the simulations.
- Chapter 2: CFD Theoretical Settings. Once the basic concepts to be taken into account in the work are clarified, it explains how the OpenFOAM program works and how it makes the necessary adjustments for the correct interpretation of the theory.
- Chapter 3: Preprocessing. In order to ensure that the results are reliable, a study on the validation of the mesh to be used is carried out. As well as the correct determination of the boundary conditions to be used.
- Chapter 4: Solver. OpenFOAM uses many different solvers and each of them uses a different algorithm. Here it is explained which is the right one according to the characteristics of the fluid in question.
- Chapter 5: Results. Last section where the results are shown and discussed. Small conclusions are highlighted for each of the data obtained.

Work Tools

In addition to OpenFOAM, a series of tools are necessary in order to proceed correctly with the investigation of the work. Those that are used are the following:

- Hardware
  - MacBook Pro, 2,6 GHz Intel Core i5, memory of 8 GB 1600 MHz DDR3.
  - University PC access. Ubuntu 18.04.1 LTS. Intel Core i7, 3.4GHz x2.
- Software
  - Ubuntu 16.04, a Linux operating system.
  - GMSH, in order to create geometries and meshes that will be used in a CFD program.
  - ParaView 5.6.0, program to visualise the results obtained in OpenFOAM. (Interface program)
  - Excel 2016, in order to handle the data extracted from ParaView.
  - Matlab R2018, in order to do the post-processing of the resulting data compiled in an Excel file. (Plots)
  - WolframAlpha, an online tool in order to calculate integrals or others equations not easily manageable.

CHAPTER 1. BASIC WIND PRINCIPLES

The wind is air in movement, a movement produced due to the different incidences of the Sun, which causes differences in air temperature, density and pressure. In the atmosphere there are concentrated areas of high pressure (anticyclone) and areas of low pressure (cyclone). These phenomena are due to the Coriolis effect and the rotation of the Earth. In the figure [1.1] can be seen a range of colours associated with a speed of wind circulation. Therefore there are areas where the air hardly moves and others where it acquires great speeds. In the anticyclones, the wind circulation does not appear due to the high pressures. On the other hand, in cyclones, the movement of air is facilitated due to low pressures.

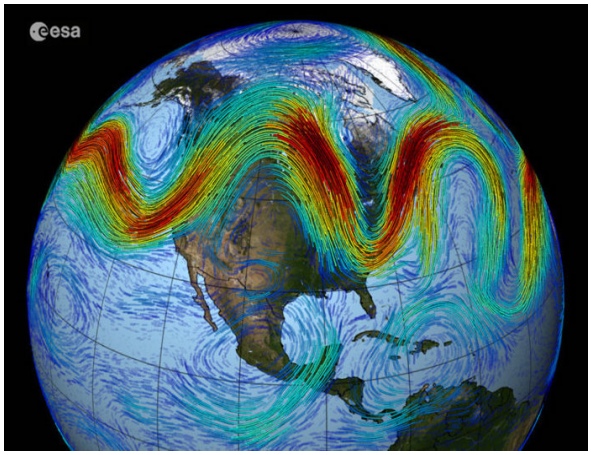


Figure 1.1: Earth's wind circulation. [1]

The air motion can appear in different layers of the Earth, in a laminar and turbulent form, although mostly turbulent. Also at high speeds depending on the areas in where is moved. Like any other fluid, it can behave differently depending on the external agents that may be encountered along its path. And it is on the surface where there is more influence, since for example a mountain can direct the wind towards narrow areas and thus accelerate its flow.

So the wind will focus on the surface layer which is the easiest layer to reach by humans and it is the layer where there is more technological activity, for wind scientific purposes. This layer is known as the *Atmospheric Boundary Layer* and it will be the area of interest throughout the work. An example of a current study on the movement of air is that of Aeolus, that is a ESA's wind mission [2]. Launched on 22 August 2018, Aeolus is the first satellite mission to acquire profiles of Earth's wind on a global scale.

1.1. Atmospheric Boundary Layer (ABL)

In this layer the aerodynamic physical conditions of the air are not constant throughout its domain. It presents turbulent behaviours depending on the physical conditions in which it is. There are external agents in the surface that cause the air to fluctuate in this layer. As for example mountain slopes, forests, cars, people, etc.

On the other hand, above this layer, there is the free atmosphere where the air is usually calmer and less turbulent. This is because *wind is approximately geostrophic (parallel to the isobars), while within the ABL the wind is affected by surface drag and*

turns across the isobars. [3]

The height at which the ABL is usually delimited is between 1km and 2km above the Earth's surface [4]. So the studies will not go beyond that height, because above it, the behaviour of the wind is stable and uniform. In addition, it is not humanly feasible to apply engineering in those heights.

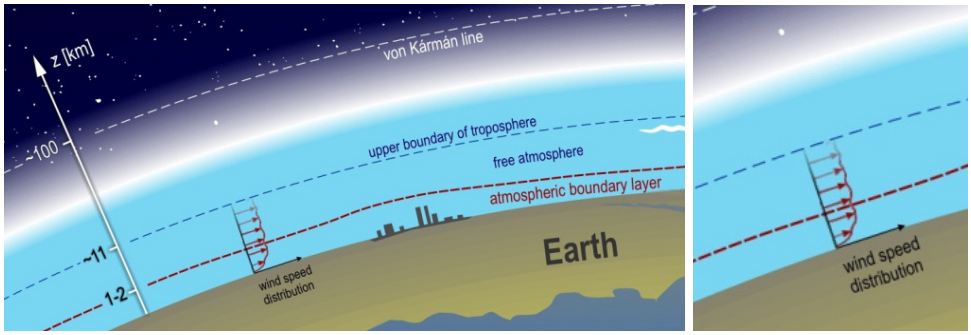


Figure 1.2: Velocity profile through different atmospheric layers. [4]

The Figure 1.2 comes from the Budapest University of Technology and Economics (BME), which will construct an atmospheric boundary layer wind tunnel in 2020 for carrying out measurements in the field of building and environmental aerodynamics [4].

In the figure, a velocity profile can be observed throughout the entire Earth's atmosphere. In the atmospheric boundary layer, a velocity gradient appears, where the speed is zero on the surface and increases logarithmically until it reaches the free atmosphere. In this layer the velocity is mostly constant and uniform up to the troposphere, which has an altitude of $z = 11km$ as it can be seen in the Figure 1.2.

It is the gradient of the velocity that will always decide the good placement of the wind turbine. Also the possible bursts and strong blows of wind that can damage it or cause an adverse effect that can not take advantage of the wind kinetic energy. So the terrain and the geographical location are important to keep it in mind.

In order to optimise the use of the wind turbine, it is necessary to know the position. It will only be possible at low altitudes, so it is necessary to study the velocity gradient. More information about the boundary layer can be found at section [1.3.].

1.1.1. Surface Roughness

Typically, due to aerodynamic drag, there is a wind gradient just a few hundred meters above the Earth's surface. Wind speed increases with increasing height above the ground, starting from zero. Flow near the surface encounters obstacles that reduce the wind speed, and introduce random vertical and horizontal velocity components at right angles to the main direction of flow. These random appearances of the components is what causes the turbulence.

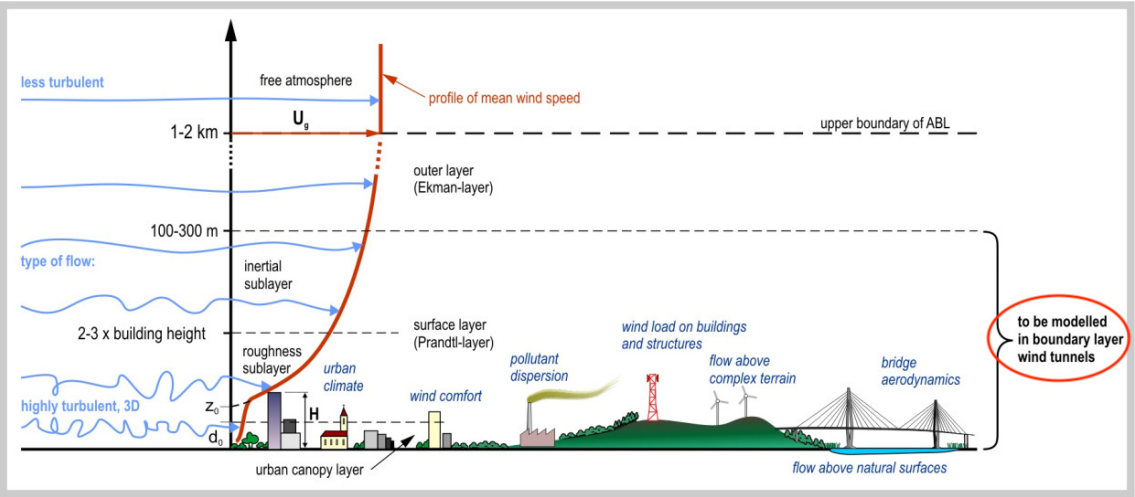


Figure 1.3: ABL wind profile throughout the roughness. [4]

The reduction in velocity near the surface is a function of roughness, so wind velocity profiles are quite different for different terrain types. Rough, irregular ground, and obstructions on the ground can reduce the geostrophic wind speed by 40% to 50%. Over open water or ice, the reduction may be only 20% to 30%.

For every kind of terrain, it is assigned a roughness length value which will affect to the intensity of the turbulence. Roughness length z_0 is the meteorological equivalent of an aerodynamic drag coefficient. So the larger the size of the obstacles, the greater the value of this parameter will be.

Experimentally, roughness lengths over many natural surfaces have been determined. Many summaries of estimated surface roughness have been prepared, with most listing only a few typical values. For example, natural surfaces, including seasonal variations, agricultural lands, urban roughness and other land categories. Here there is an example:

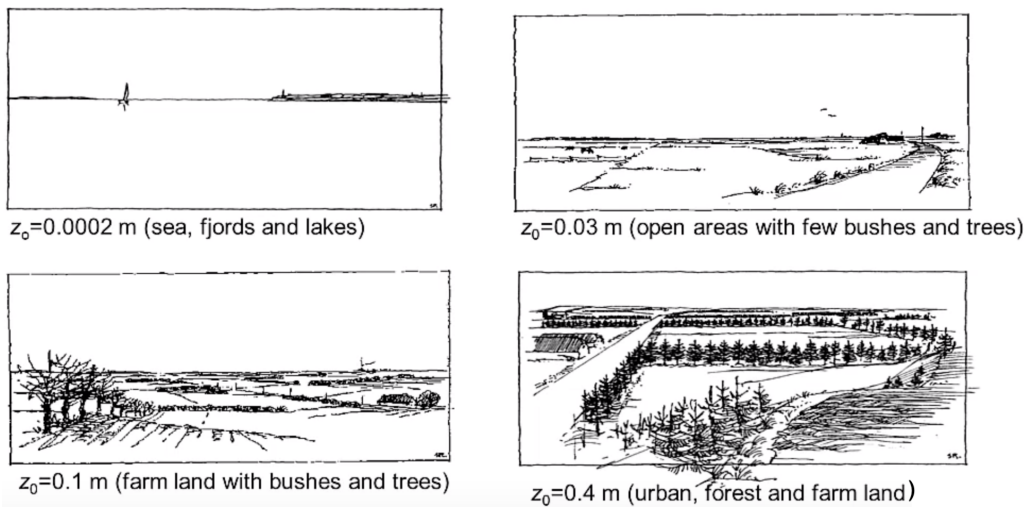


Figure 1.4: Different roughness z_0. [5]

A typical table on different types of terrain and their equivalent roughness length values can be seen below:

Terrain types	Roughness length $z_0(m)$
Cities, forests	0.7
Suburbs, wooded countryside	0.3
Villages, countryside with trees and hedges	0.1
Open farmland, few trees and buildings	0.03
Flat grassy plains	0.01
Flat desert, rough sea	0.001

Table 1.1: Terrain types, roughness length. [6]

As it can be seen, the greater the air obstruction on the surface, the greater the roughness length. Keep in mind that this does not represent the height of the present objects, but an experimental data that will be part of the ABL equations, within logarithmic profiles.

1.2. General Aerodynamic Equations

Nowadays the behaviour of the wind is difficult to define mathematically, since it does not move with simple patterns. Newton's equations and those of Naiver-Stokes, have been able to simulate remarkably the movement of air or any other fluid. They are the main basic references to apply in any problem in fluid mechanics.

In this section, the general aerodynamic equations of the fluid are detailed, depending on its theoretical hypothesis adapted to the purposes of the atmospheric boundary layer.

1.2.1. Fluid hypothesis

In order to approach the study to a merely behaviour of the atmospheric boundary layer, it has been decided to take the following characteristics of the fluid:

- Incompressible, so the material density is constant within a fluid parcel. $(\delta\rho/\delta t = 0;\ \delta\rho/\delta x = 0;\ \ \delta\rho/\delta y = 0;\ \delta\rho/\delta z = 0)$
- Steady-state, so the state variables which define the behaviour of the system are unchanging in time. $(\delta/\delta t = 0)$
- Turbulent. Chaotic changes in pressure and flow velocity because of the roughness in this case.
- Single phase. Because we are working with air, so in this case we have a single homogeneous phase that is the steam.
- Isothermal, so the fluid temperature is considered to be constant. $(\delta/\delta T = 0)$

At $P_{atm} = 101.325kPa$ and $T_{atm} = 15^oC$ according to ISA (International Standard Atmosphere) [7]:

- $\rho = 1.225kg/m^3$ (Density)
- $\mu = 1.802 \cdot 10^{-5}kg/ms$ (Dynamic Viscosity)
- $\nu = 1.47 \cdot 10^{-5}m^2/s$ (Kinematic Viscosity)

Bearing this in mind, the main equations of the fluid mechanics will be reduced and the computation will be fewer when these features are applied to OpenFOAM. So according to the reference of the *Fundamentals of Aerodynamics by John D. Anderson, Jr.* [8], it can be found the Continuity equation and the Momentum conservation equation.

The Energy equation does not intervene because the fluid is isothermal ($\delta/\delta T = 0$).

1.2.2. Continuity equation

Starting from the explanations defined in the Anderson [8], it can be got the following equation for continuity:

$$\oiiint_V \left[\frac{\delta\rho}{\delta t} + \nabla(\rho\vec{V})\right] dV = 0 \tag{1.1}$$

$$\boxed{\frac{\delta\rho}{\delta t} + \nabla(\rho\vec{V}) = 0}$$

The equation (1.1), is the continuity equation in the form of a partial differential equation. This equation relates the flow field variables at a point in the flow and not in a finite space.

The equations are going to be used in a CFD program so they shall be as simple as possible in order to be able to interpret them in a matrix form. In OpenFOAM, all are expressed in differential form since all their solvers are defined by ODEs (*Ordinary Differential Equation (ODE) solvers*).

Keeping in mind the hypothesis of the fluid [1.2.1.] and its characteristics mentioned above, the equation of differential form are as follows:

- Differential formulation

$$\frac{\delta\rho}{\delta t} + \nabla \cdot (\rho\vec{V}) = \frac{D\rho}{Dt} + \rho\nabla \cdot \vec{V} = 0 \tag{1.2}$$

- Simplification

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \vec{V} = 0$$

$$\frac{D\rho}{Dt} = \frac{\delta\rho}{\delta t} + \vec{V} \cdot \nabla\rho = 0$$

$$\nabla \cdot \vec{V} = 0 \tag{1.3}$$

The speed of a fluid in motion forms a vector field. Each particle of the fluid moves with a certain speed, in a certain direction. Logically the number of particles that enter a volume is the same as the particles that come out of that volume. That is to say that the divergence of the velocity field of an incompressible fluid is zero.

1.2.3. Momentum conservation equation

- Differential formulation

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \nabla \cdot \tau' + \rho \vec{F_m} \tag{1.4}$$

- Simplification

$$\frac{D\vec{V}}{Dt} = -\frac{1}{\rho}\nabla p + \nabla \cdot \nu + \vec{F_m} \tag{1.5}$$

Because in the OpenFOAM simulations it is used the pressure divided by the density and not the pressure only, so it is better this expression. ($p = \frac{1}{\rho}P$) [$\frac{m^2}{s^2}$]. And in the diffusion term we have the kinematic viscosity(ν) instead of the dynamic viscosity(μ).

1.3. Boundary Layer Concept

Within the boundary layer the fluid can behave laminarly, turbulently and then there is the transition phase between both states.

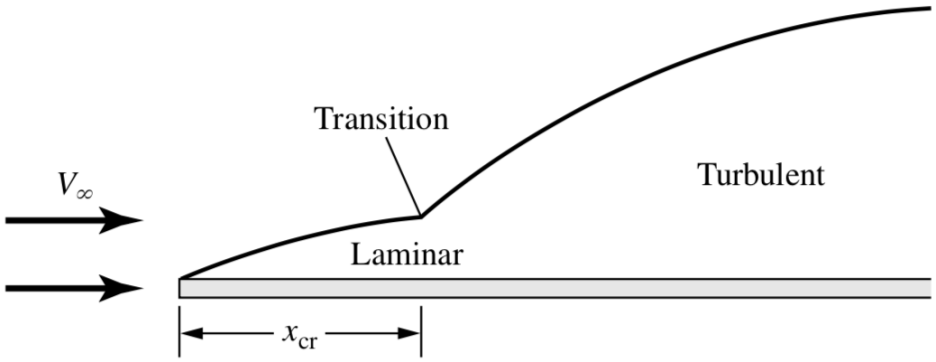


Figure 1.5: Transition from laminar to turbulent flow. [8]

The transition from laminar to turbulent flow takes place over a finite region, as sketched in Figure 1.5. However, for purposes of analysis, it is usually considered the transition region as a single point, called the critical transition point, upstream of which the flow is laminar and downstream of which the flow is turbulent. The distance of the transition point is denoted by x_{cr}. The value of x_{cr} depends on a whole host of phenomena. For example, some characteristics which encourage transition from laminar to turbulent flow, and hence reduce x_{cr}, are:

- Increased surface roughness. Indeed, to promote turbulent flow over a body, rough grit can be placed on the surface near the leading edge to trip the laminar flow into turbulent flow.
- Adverse pressure gradients. In addition to causing flow-field separation, an adverse pressure gradient strongly favours transition to turbulent flow.
- Heating of the fluid by the surface. If the surface temperature is warmer than the adjacent fluid, such that heat is transferred to the fluid from the surface, the instabilities in the laminar flow will be amplified, thus favouring early transition. But as in this case the temperature exchange is not taken into account, this factor does not intervene.

The Reynolds number itself is a dominant factor in transition to turbulent flow. If the critical transition point is considered, then the Reynolds number can be defined as:

$$Re_{cr} \equiv \frac{\rho_\infty \cdot U_\infty \cdot x_{cr}}{\mu_\infty} \tag{1.6}$$

According to the *Fundamentals of Aerodynamics by John D. Anderson, Jr.*, the value of Re_{cr} for a given body under specified conditions is difficult to predict; indeed, the analysis of transition is still a very active area of modern aerodynamic research. As a rule of thumb in practical applications, it is frequently $Re_{cr} \approx 5 \cdot 10^5$ [8]; if the flow is below this value, it will be laminar, and if the value is much larger, then the flow is most likely turbulent.

In the event that a rough surface intervenes in the fluid, by applying the mass conservation concept ($Q_\infty = U_\infty \cdot A$), where Q_∞ is the volumetric flow rate (m^3/s) and A is the cross-sectional area of the domine (m^2), the Reynolds equation can be considered as follows:

$$Re_{cr} \equiv \frac{\rho_\infty \cdot Q_\infty \cdot x_{cr}}{\mu_\infty \cdot A} \tag{1.7}$$

Roughness interferes with the transverse flow path through the domain.This causes the cross-sectional area to be reduced. And the smaller the area, the higher the Reynolds number. So the more roughness the more turbulence you get.

1.3.1. Turbulence

The turbulence or turbulent flow is fluid motion characterised by chaotic changes in pressure and flow velocity. Below there is an example:

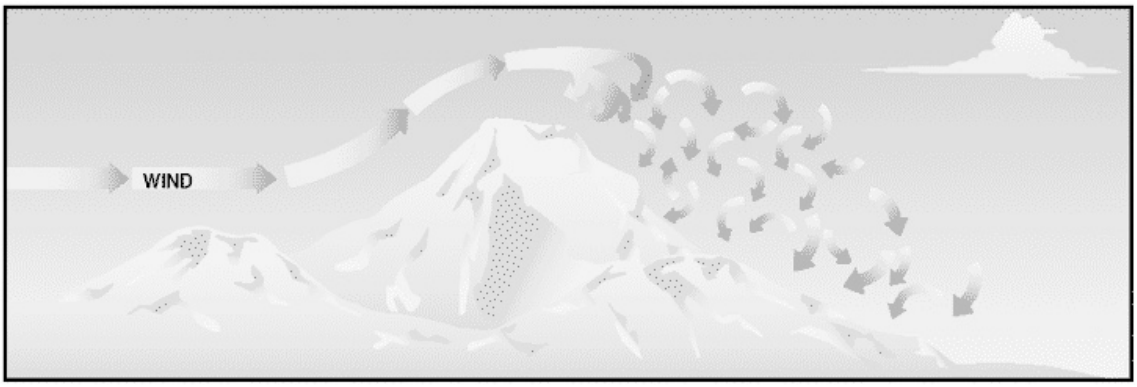


Figure 1.6: Mountain wave. [9]

It can be seen how the wind meets a mountain as an obstacle. This flow is forced to climb the slope until it reaches the top, where it meets a pressure difference that forces the wind to descend rapidly. This pressure difference causes the fluid to behave turbulently. This concept is known as Mountain Wave and is usually a problem for aviation.

There are many ways to model turbulence analytically. Three large groups can divide some turbulence models: RANS, DES and LES. And within the RANS models it can be found: k-epsilon, k-Omega, SST,SAS SST, etc. But the CFD OpenFOAM program uses the k-epsilon (RANS group) model to represent the turbulence of atmospheric air. It is the most common model used in Computational Fluid Dynamics (CFD) to simulate mean flow characteristics for turbulent flow conditions. [10]

1.3.2. Standard $k-\epsilon$ turbulence model

In 1974, Launder and Spalding [11] proposed a practical approach for the $k-\epsilon$ turbulence model. This is the model that Richard and Hoxey took as reference in order to simplify their values of k and epsilon.

The turbulence kinetic energy (k), is the accumulated energy of the fluid turbulence. The rate of dissipation of turbulence energy (ϵ), is the scale at which turbulence kinetic energy is converted into thermal internal energy dissipated. The main theoretical equations are then distinguished:

For turbulent kinetic energy k [12]

$$\frac{\delta(\rho k)}{\delta t}+\frac{\delta(\rho k u_i)}{\delta x_i}=\frac{\delta}{\delta x_j}[\frac{\mu_t}{\sigma_k}\frac{\delta k}{\delta x_j}]+2\mu_t E_{ij}E_{ij}-\rho\epsilon \tag{1.8}$$

For dissipation ϵ [12]

$$\frac{\delta(\rho \epsilon)}{\delta t}+\frac{\delta(\rho \epsilon u_i)}{\delta x_i}=\frac{\delta}{\delta x_j}[\frac{\mu_t}{\sigma_\epsilon}\frac{\delta \epsilon}{\delta x_j}]+C_{1\epsilon}\frac{\epsilon}{k}2\mu_t E_{ij}E_{ij}-C_{2\epsilon}\rho\frac{\epsilon^2}{k} \tag{1.9}$$

where:

- u_i represents velocity component in corresponding direction.
- E_{ij} represents component of rate of deformation. This value is the deformation of a material with respect to time. It has two different coefficient: the expansion and the shear rate. The rate of deformation is linearly proportional to the viscous stress of a Newtonian fluid.
- μ_t represents eddy viscosity:

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \tag{1.10}$$

The equations also consist of some adjustable constants $\sigma_k, \sigma_\epsilon, C_{1\epsilon}$ *and* $C_{2\epsilon}$. The values of these constants have been arrived at by numerous iterations of data fitting for a wide range of turbulent flows. These are as follows:

$$C_\mu = 0.09;\ \ \sigma_k = 1.00;\ \ \sigma_\epsilon = 1.30;\ \ C_{1\epsilon} = 1.44;\ \ C_{2\epsilon} = 1.92$$

And the von Karman constant can be defined by these constants by:

$$\kappa = \sqrt{(C_{2\epsilon} - C_{1\epsilon})\sigma_\epsilon \sqrt{C_\mu}} \approx 0.4 \tag{1.11}$$

The Karman constant adopts a significant importance in the values of k and epsilon that propose Richard and Hoxey in their ABL model equations, because it depends on the adjustable constants of the standard $k-\epsilon$ turbulence model. This ABL model equations will be explained below.

1.4. Richard and Hoxey's ABL model

The most successful modelling of the behaviour of the neutral Atmospheric Boundary Layer as an *horizontally-homogeneous turbulent boundary layer*, directed to CFD programs, is the one proposed by Richard and Hoxey. They were able to conclude with simplicity a velocity profile and $k-\epsilon$ turbulence model in 1993 [13].

In an attempt to provide guidance for accurately represent the ABL flow, Richards and Hoxey recommended modelling the ABL as a horizontally homogenous turbulence surface layer, and a set of inlet profiles of mean wind speed and turbulence quantities for the standard $k-\epsilon$ model was proposed. Two inlet boundary conditions are defined thanks to the analytical approaches they made on the ABL.

1.4.1. Non-uniform inflow conditions

Mathematically the terms of velocity and turbulence come from the general aerodynamic equations such as the continuity or momentum one. But above all, these ABL adjustments have been made thanks to the experimental tests that they did. This boundary condition consists of introducing the theoretical equations of the ABL, which represent the gradient of velocities when it meets a rough surface, to the inlet of the

computational domain. That is, what is introduced is directly the theoretical logarithmic profile that represents the velocity gradient that occurs in the Atmospheric Boundary Layer.

There is a factor inside the velocity equation that is omitted in this work, the stability factor Ψ. This factor shifts the profile depending on whether we are in the morning or at night, but it has no relevant effect for a standard atmospherical model [6]. They propose then the following equations:

$$U(z) = \frac{U^\star}{\kappa}[ln(\frac{z+z_0-z_g}{z_0}) + \cancel{\Psi}] \tag{1.12}$$

$$k(z) = \frac{(U^\star)^2}{\sqrt{C_\mu}} \tag{1.13}$$

$$\epsilon(z) = \frac{(U^\star)^3}{\kappa(z+z_0-z_g)} \tag{1.14}$$

Where:

- z, represents all the points along the vertical axis. (Height)
- z_0, is the length roughness. [m]
- z_g, is the minimum z-coordinate [m]
- κ, is the von Karman's constant. ($\kappa \approx 0.4$)
- C_μ, is a constant of the $k-\epsilon$ turbulence model.
- U_{ref} [m], is the velocity that we find at a reference height Z_{ref} [m] imposed by the user.
- $U^\star$, is the friction velocity.

The friction velocity is defined as follows:

$$U^\star = \frac{\kappa \cdot U_{ref}}{ln(\frac{Z_{ref}+z_0-z_g}{z_0})} \tag{1.15}$$

As it has been said before, these values for k and epsilon are indicative and simplified, since the turbulence equations can be represented differently depending on the values of their *adjustable constants*. Constants that have been explained in the section [1.3.2.].

As roughness is a factor causing turbulence, what is wanted is to reduce other factors that may increase it. That is why it is decided to impose an entry speed of $U_{ref} = 1m/s$. This speed is very low and therefore reduces the chances of turbulence appearing.

Since the input speed in this case is not uniform, it will be highlighted that this velocity is always at a height of $Z_{ref} = 50m$.

1.4.2. Uniform inflow conditions

On the other hand, there is a constant inlet condition. It consists of taking the ABL equations (1.12) (1.13) (1.14) of Richard and Hoxey and setting values in order to guarantee that the velocity and turbulence are uniform on the z-axis. That means that it will have the same velocity when $z = z_{top}(m)$ and when $z = \alpha(m)$, where 'α' is the distance between the wall and the first node of the cell. Because inside the no-slip condition at the bottom, means that when $z = 0(m)$ there will be no speed $U(0) = 0(m/s)$. The cell and the grid will be explained in [CHAPTER 3] of the work. So the equations would be:

$$U(z) = \frac{U^\star}{\kappa} ln(\frac{Z_{ref} + z_0 - z_g}{z_0}) = U_{constant} \tag{1.16}$$

$$k(z) = \frac{(U^\star)^2}{\sqrt{C_\mu}} \tag{1.17}$$

$$\epsilon(z) = \frac{(U^\star)^3}{\kappa(Z_{ref} + z_0 - z_g)} = \epsilon_{constant} \tag{1.18}$$

And they will be the initial conditions of the simulations. As well as in the non-uniform inflow conditions [1.4.1.], the inlet velocity along the z-axis will be $U_{constant} = 1m/s$.

CHAPTER 2. CFD THEORETICAL SETTINGS

In this chapter it is explained the adjustments that OpenFOAM applies in order to interpret the theoretical concepts in the simulations: solver to use, boundary conditions, modelling the equations as well as the modelling of the turbulence with the wall functions.

2.1. Turbulence modeling

When applying the roughness in the lower part of the geometry, OpenFOAM recommends the use of the wall functions. This method of calculation allows to obtain good results without the need to use a fine mesh and, in this way, it would save computational cost. But it must be taken into account that the coarse mesh doesn't accurately define the k-epsilon gradients nor for velocity. So it is important to know the performance of the wall functions and how it works.

Once the turbulence analytical model has been explained, it must now see how OpenFOAM models these equations in a matrix form in order to apply it in the c_{++} language. Notice that the following modelings are used in the wall functions of the OpenFOAM.

2.1.1. Transport equation for k

In order to simplify the analytical model of the equations (1.8) and (1.9), OpenFOAM divides everything by the density, obtaining then the turbulent viscosity ($\nu_t = \mu_t/\rho$) and more understandable differentials:

- Original Equation

$$\frac{\delta k}{\delta t} + \vec{u_i}\frac{\delta k}{\delta x_i} = \nabla \cdot (\frac{\nu_\tau}{\sigma_k}\nabla k) + P - \varepsilon \tag{2.1}$$

- OpenFOAM modelling approach [14]

$$\frac{\delta k}{\delta t} + \nabla \cdot (Uk) - (\nabla \cdot U)k - \nabla D_{k,eff}\nabla k = G - \varepsilon \tag{2.2}$$

The term $(\nabla \cdot U)k = 0$ because it is working with an incompressible fluid. And the diffusion term is $D_{k,eff} = \nu + \nu_T$. It can be seen that the matrix approximation it's nearly the same as in the original equation because:

$$\begin{aligned} \nabla \cdot (Uk) &= U \cdot \nabla k + (\nabla \cdot U)k \\ U \cdot \nabla k &= \nabla(Uk) - (\nabla \cdot U)k \\ \vec{u_i}\frac{\delta k}{\delta x_i} &= U \cdot \nabla k \end{aligned}$$

And $G = P$ that is $P = 2\nu_t E_{ij} E_{ij}$

2.1.2. Transport equation for ϵ

- Original Equation

$$\frac{\delta\epsilon}{\delta t} + \vec{u_i}\frac{\delta\epsilon}{\delta x_i} = \nabla\cdot(\frac{\nu_t}{\sigma_\epsilon}\nabla\epsilon) + C_{\epsilon 1}\frac{P\varepsilon}{k} - C_{\epsilon 2}\frac{\varepsilon^2}{k} \tag{2.3}$$

  The epsilon transport equation has the same expression of the kinetic energy transport equation but with two different production and dissipation term.

- OpenFOAM modelling approach [14]

$$\frac{\delta\epsilon}{\delta t} + \nabla\cdot(U\epsilon) - (\nabla\cdot U)\epsilon - \nabla D_{k,eff}\nabla\epsilon = C_1 G\frac{\varepsilon}{k} - C_2\varepsilon\frac{\varepsilon}{k} \tag{2.4}$$

  And the diffusion term is $D_{k,eff} = \nu + \frac{\nu_t}{\sigma_\epsilon}$.

  Once OpenFOAM has the values of k and epsilon, the value of the turbulent viscosity can be calculated: (1.10)

$$\nu_t = C_\mu\frac{k^2}{\epsilon} \tag{2.5}$$

  The value of the turbulent viscosity is the value which will describe the friction near the wall. So the Reynolds stresses can be defined like:

$$-\overline{(u_i' u_j')} = \nu_t(\frac{\delta\overline{u_i}}{\delta x_j} + \frac{\delta\overline{u_j}}{\delta x_i}) - \frac{2}{3}k\delta_{ij} \tag{2.6}$$

2.2. Wall function and y^+ concept

In 1972, Tennekes and Lumley [15] described a series of analytical solutions for surface flows in a channel, pipe or even other boundary layers with different Reynolds numbers. These solutions can be used for complex flows such as flows with various pressure gradients, zero wall stress and rough surfaces. In this case a roughness is studied and carried it out in order to represent the behaviour of the Atmospheric Boundary Layer (ABL). They are used, basically, to bridge the near-wall region of turbulent flows so that there is no need to have the fine grids near the wall. Unless the near-wall flow structures are required to resolve. These solutions are referred to as the *wall functions*. Furthermore, a generalised and unified law of the wall which is valid for surface layer (including *viscous sublayer*, *buffer layer* and *inertial sublayer* or *Log-law region*) is analytically constructed, taking as reference the "A Generalised Wall Function" review [16] in 1999.

The first step in applying wall functions is to compute the friction velocity and the wall shear stress. The friction velocity is then used to set the boundary conditions for k and ϵ at the grid point adjacent to the wall. Finally, the wall shear stress is used in the computation of the diffusion term in the Navier-Stokes equations at the grid point adjacent to the wall.

An appropriate velocity scale for flow in the near-wall region is the friction velocity, defined by:

$$U^{\star} \equiv \sqrt{\frac{\tau_w}{\rho_w}} \tag{2.7}$$

Where τ_w ($\frac{kg}{m \cdot s^2}$) is the wall shear stress and ρ_w ($\frac{kg}{m^3}$) is the density at the wall. This would be the origin equation (1.15) from which Richard and Hoxey departed to find the friction velocity defined in the previous chapter. The equation (2.7) is the general mathematical expression of the friction velocity, but the equation that will be used is the modelled one. Then the value of the friction velocity will depend on the roughness length. If it is defined a roughness length between the values of $z_0 = [1cm, 100cm]$, the resulting U^* values:

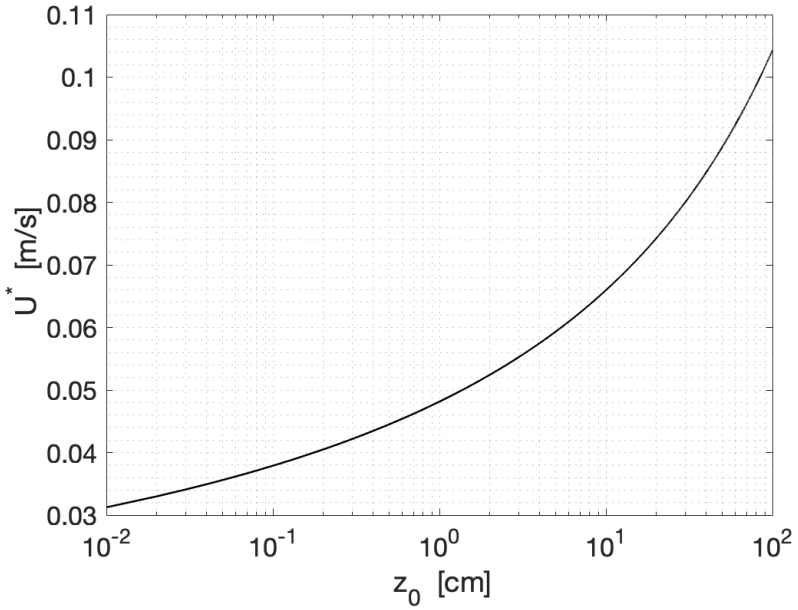


Figure 2.1: Roughness z_0 vs friction velocity U^*.

Therefore the more roughness, more friction velocity and therefore more brake speed. In order to simplify values and data, the parameters are non-dimensioned, in this way values can be standardised. Using this velocity scale, a non-dimensional velocity and a non-dimensional length are defined by:

$$u^{+} \equiv \frac{u}{U^{\star}} \tag{2.8}$$

and

$$y^{+} \equiv \frac{U^{\star} \cdot \alpha}{\nu} \tag{2.9}$$

Where "u" (m/s) is the velocity component parallel to the wall, "α" (m) is the distance normal to the wall, and "ν" is the kinematic viscosity. Since the value of y^+ and u^+ depend on the distance between the first node of the mesh with respect to the ground, the values will be determined in the preprocessing chapter, in the table [3.3], where the study of the validation of the mesh is found.

The equation which describes the velocity profile in the log region is:

$$u^+ = \frac{1}{\kappa} ln y^+ + B \tag{2.10}$$

Where κ is the von Karman constant and B is an aditional constant. Values of κ and B have been empirically determined to fall in the ranges 0.40-0.41 and 4.9-5.5 respectively (Cebeci and Smith 1974 [17]). In the viscous sublayer,

$$u^+ = y^+ \tag{2.11}$$

Notice that te equations describing the velocity profile in the inner region are collectively called the "law of the wall." Imposing a minimum distance of $\alpha = 1mm$, using the values determined above and take the equation (2.9), it can be obtained the following expression:

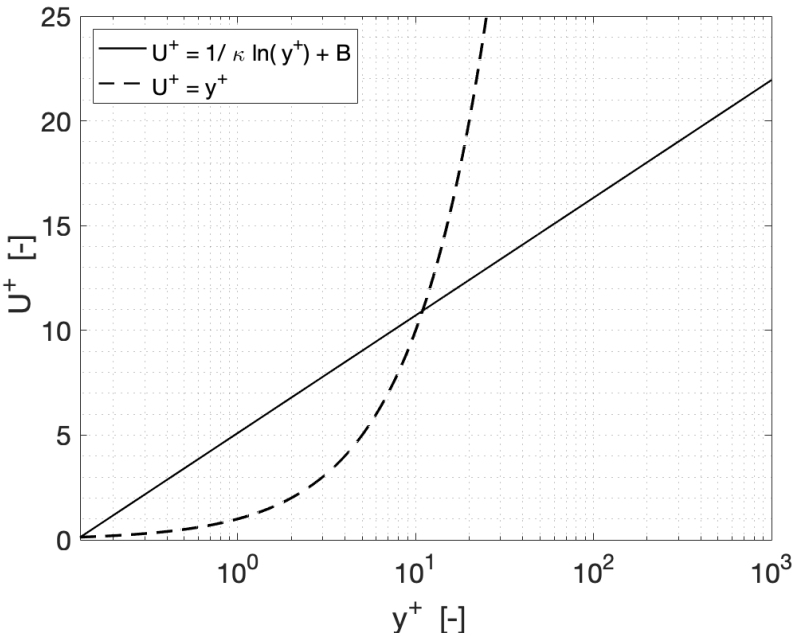


Figure 2.2: Standard y^+ vs u^+.

These are the two equations that the wall functions take into account to get the velocity profile close to the ground. The next section specifies the problem of the transition between the viscous sublayer and the log region.

2.2.1. Turbulence dumping problem

Wall function equations for k-epsilon turbulence present better results for a high value of Reynolds. The problem that exists in the wall functions is related to the value of C_μ.

C_μ is taken as a constant value once the turbulence have adopted a value, after having separated from the wall. It must be remembered that in the wall there is no velocity so there is no turbulence and k and epsilon would not exist. It is C_μ that causes an

abrupt variation in the values of k and epsilon since before being constant it obtains the following expression:

$$C_\mu(y) = \frac{-\overline{(u_i' u_j')}}{\sqrt{u_i'^2}\sqrt{u_j'^2}} \tag{2.12}$$

Where y belongs to the points very close to the wall.

This is the reason that in the wall functions you don't get good fitting in the velocity, because there are abrupt changes in the k-epsilon values at the beginning of the fitting. This section is denominated as a Buffer Layer. There are three different layers depending on the y^+ values:

- Viscous sublayer $(y^+ < 5)$

$$u^+ = y^+$$

- Buffer layer $(5 < y^+ < 30)$

$$u^+ \neq y^+$$

$$u^+ \neq \frac{1}{\kappa} ln y^+ + B$$

- Log-law region $(30 < y^+ < 300)$

$$u^+ = \frac{1}{\kappa} ln y^+ + B$$

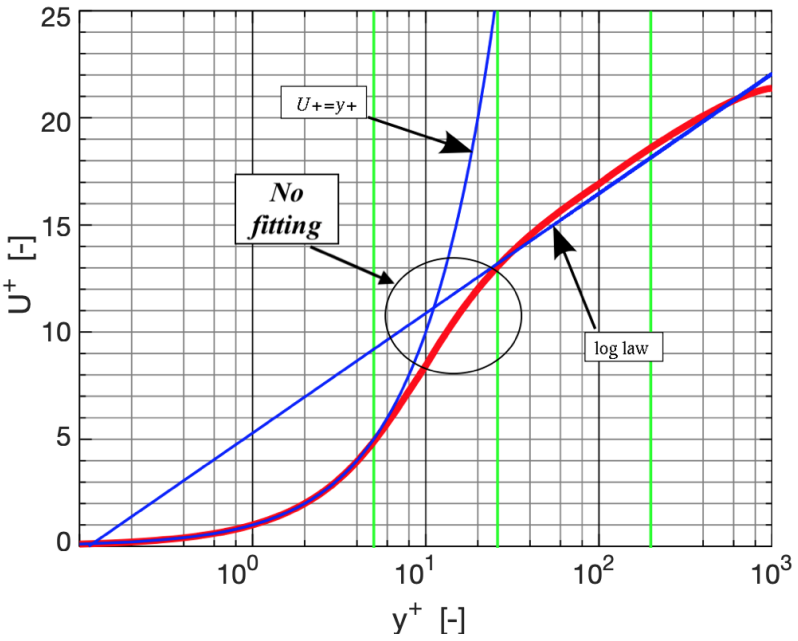


Figure 2.3: Velocity fitting produced by wall functions. [19]

In red it can be seen the velocity produced by the wall functions of any CFD program.

Keep in mind that the y^+ nomenclature, is denominated with the α letter for scientific reasons or for scientific customs of definition. But this dimensionless value is related with the vertical axis, that is, the Z axis in this case.

In order to have good fitting at the bottom, it is important to avoid the Buffer Layer, and this is only possible taking a mesh inside the log-law region. But the inconvenient of this conclusion, is the bad accuracy of the final results of the simulation. Using a coarse mesh won't guarantee good fitting in the rest of the computational domine. Therefore the user will determine the mesh according with its priorities. If the Buffer Layer appears in a not important region, the user can go ahead regardless of it.

2.3. Physical Surfaces

In OpenFOAM, each boundary condition has its characteristic objective. In this section, it will be explained those that are going to be used throughout the work and those essential to achieve the study that is intended. For example the wall functions, which will guarantee the simulation of the roughness of the terrain; the ABL functions, which will ensure the turbulent behaviour of the wind. And other functions that will be described.

The boundary conditions that are used for the study of the convergence of the mesh, will be simply the following:

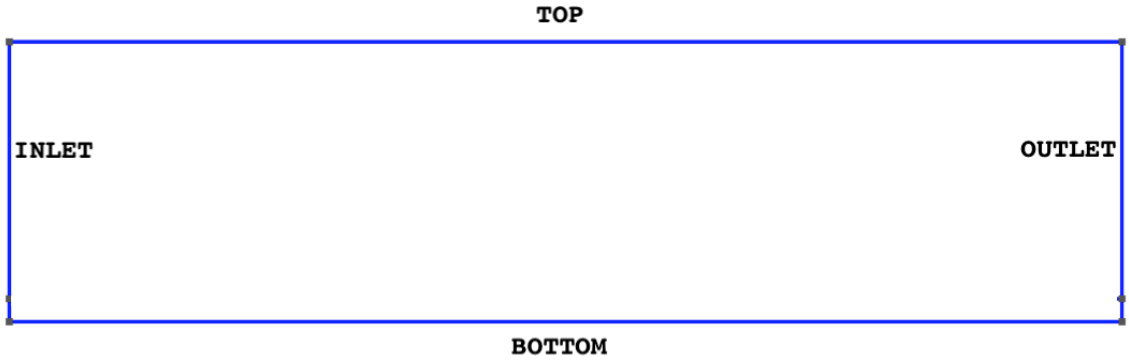


Figure 2.4: Physical surfaces in the basic geometry.

	Inlet	Outlet	Top	Bottom	Front & Back
$U_z(m/s)$	fixedValue	zeroGradient	slip	noSlip	symmetry
$P_z(m^2/s^2)$	zeroGradient	fixedValue	zeroGradient	zeroGradient	symmetry
k	atmBoundaryLayerInletK	inletOutlet	slip	kqRWallFunction	symmetry
ϵ	atmBoundaryLayerInletEpsilon	inletOutlet	slip	epsilonWallFunction	symmetry
ν_t	calculated	calculated	slip	nutkAtmRoughWallFunction	symmetry

Table 2.1: Different OpenFOAM boundary conditions (used for the mesh validation).

Notice that in the pressure, OpenFOAM determines that physical magnitude like the pressure divided by the density. So the units will be m^2/s^2. In the files, everything is divided by the density in order to simplify matrix equations. The definition of every boundary condition is defined as follows:

- *fixedValue* boundary condition allows to fix an uniform value in that domain of a physical magnitud like velocity, pressure, temperature, etc.
- *zeroGradient* boundary condition simply extrapolates the quantity to the patch from the nearest cell value. The meaning is, the quantity is developed in space and its gradient is equal to zero in direction perpendicular to the patch (perpendicular to the boundary).
- The *inletOutlet* boundary condition is normally the same as zeroGradient, but it switches to fixedValue if the velocity vector next to the boundary aims inside the domain (backward flow). The value of that fixedValue is inletValue.

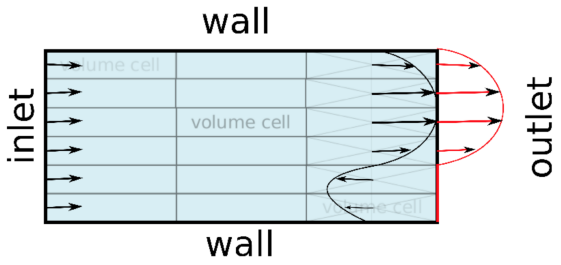


Figure 2.5: inletOutlet at outlet patch. [20]

- The *slip* boundary condition erases the normal component of the variable at the patch and keeps the tangential components untouched. It is very similar to the symmetry. Keep in mind that the symmetry is a boundary type, whereas the slip is a boundary condition which can be applied on boundary patches of type wall and patch respectively.
- The *noSlip* boundary condition is an alternative to the zero fixedValue boundary condition for velocity. There is no difference between them. Means that the velocity is 0 m/s in that boundary.

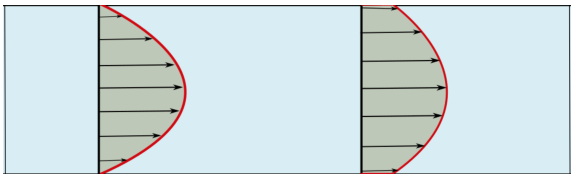


Figure 2.6: noSlip vs Slip condition. [20]

- The *symmetry* boundary condition enforces a symmetry constraint. It makes a mirror of the quantity in that point of the mesh.
- The *calculated* boundary condition is not designed to be evaluated; it is assumed that the value is assigned via field assignment, and not via a call to e.g. update coefficients or evaluate. It depends on the other quantities.
- The *atmBoundaryLayerInletK* and the *atmBoundaryLayerInletEpsilon* boundary conditions, are those that behave as a function on that physical magnitude, which in that file it is composed a series of formulas to calculate the component in question at a given value like height or time for example.

- The *kqRWallFunction*, *epsilonWallFunction* and *nutkAtmRoughWallFunction* are wall functions. In them are all the formulas and conditions that have been highlighted in the law of the wall section. (2.2.)

Later on, the boundary conditions will be determined, since there are several possibilities in the Inlet and in the Top.

CHAPTER 3. PREPROCESSING

Before starting with the simulations, it is necessary to find a reliable mesh that is able to interpret the flow of the fluid. Also it is essential to find the boundary conditions so that the initialisation, the development and the outcome of the simulation are carried out correctly.

3.1. Geometry and Mesh

In order to verify that OpenFOAM is able to represent results similar to those that Richard and Hoxey concluded after a long study, what is proposed is a simple rectangular geometry to represent an inlet, an outlet, a rough terrain, the front and back boundaries, and a top layer that will control the fluid in that volume.

Let's see the velocity profile through the path along the horizontal X axis and along the vertical Z axis, in the control volume the axis in depth Y, will not be highlighted. The last third dimension is created to bring the study closer to reality, but it will not intervene in the modification of the behaviour of the fluid after all. Therefore, it may be neglected once it has been verified that the velocity profile does not vary along the Y axis. This affirmation will be verified in the results [CHAPTER 5].

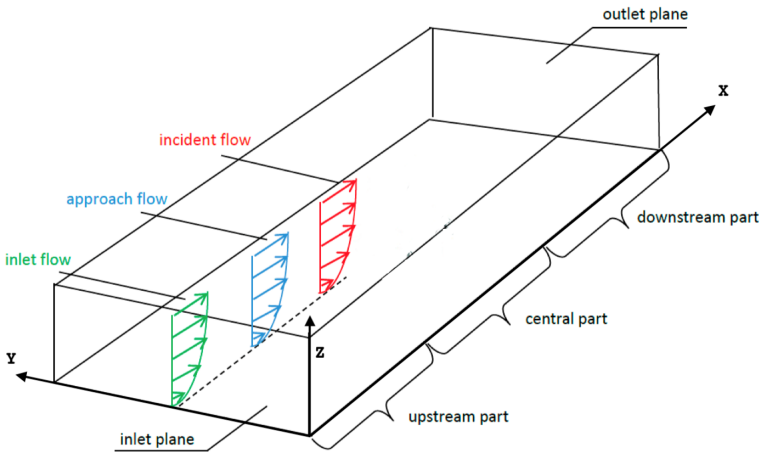


Figure 3.1: Example of the control volume proposed. [27]

Then, it is proposed a severe height of 50 meters, a depth of 5 meters, and a length of 200 meters of travel, to see the evolution of the velocity profile through the roughness.

The GMSH program has been used for the creation of the geometry, since it is a program compatible with OpenFOAM and there are no compatibility problems between them. Also this program is capable of generating meshes within that geometry, mesh that can be transformed into c_{++} files so that it can be read by OpenFOAM. In this way, a large workload is avoided with OpenFOAM files, since GMSH easily generates them once the mesh is designed from a visual interface.

The geometry is presented as follows:

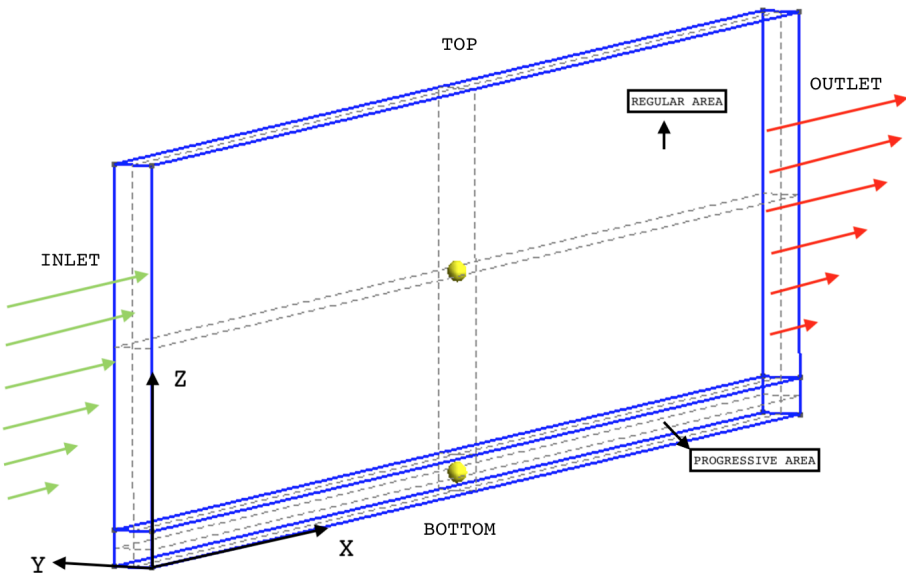


Figure 3.2: Geometry in GMSH.

It is differenced two regions because the roughness is presented close to the wall and it is necessary to highlight a progression area. Means that in that region of the rectangle it will be generated a progressive mesh, which will start with a very small separation distance between nodes at the bottom and will end with a greater distance until reaching the regular zone, which means that in that section, the distance between nodes will be the same. The main objective of this section is to create a mesh that nor require much calculation time.

To obtain more precision in the behaviour of any fluid in CFD, what is done is to generate a mesh with many calculation points. In that way, the program can apply more physical equations in less area distance and ensure analytical results. But the drawback of making a very fine mesh, is that it would take a long time to find a solution or result. Result, which could be obtained without creating a super fine mesh.

In CFD, the nodes are usually concentrated in those parts where it is known that there may be more turbulence or more physical gradients. That way you would get more accuracy in the gradient. But in areas where these turbulences are dispensable, it is not necessary to create a mesh with many nodes since the behaviour of the fluid is practically stable throughout that region. And that is why the progressive zone is created in the geometry, since the roughness in the ground will generate more turbulence at the bottom.

3.1.1. Different Meshes Proposed

The results obtained in a simulation can be displayed depending on the mesh. As has been said before, in CFD what is always desired, is to obtain reliable and good results through meshes that require little calculation time. But this is not easy to achieve on the first attempt, unless you have much experience and knowledge on the subject.

An ideal way to know when a mesh is accurate without the need to introduce more elements, is to take any boundary conditions, make them fixed and go simulating with different meshes. The idea is to gradually increase the number of elements until the solution that is obtained, does not vary significantly with respect to the mesh. That is, the solution converges independently of the number of nodes added to it.

Therefore 5 different meshes with different separate distances in the regular zone are proposed:

Mesh	dz (m)
1	0.15
2	0.20
3	0.25
4	0.30
5	0.40

Table 3.1: Different meshes with different separations between elements.

The progressive zone will adapt to the regular zone, being the element separation of the progressive zone shorter than the other. The zone separation height h will remain constant at $z = h = 5m$, as well as the distance between the bottom and the first node of the cell ($\alpha \approx 1mm$). The only thing that will change, will be the number of nodes n and the ratio r, that will decide the increase of the separation per node. According to the *Digital Library of Mathematical Functions* [18], a mathematical expression can determine the relationship between a distance h and the number of nodes n that contains that distance, applying an increase progression ratio r:

$$h = \frac{\alpha(1 - r^n)}{1 - r} \tag{3.1}$$

In order to adapt both regions, it is important to equalise this expression with the one that can compute the last separation node:

$$\alpha_n = \alpha r^{n-1} \tag{3.2}$$

Where α_n is the last and the biggest separation distance between nodes in the height h in this case. The idea is to assign the value of α_n to the regular distance of every mesh ($\alpha_n = 0.15m$, $\alpha_n = 0.20m$, $\alpha_n = 0.25m$, ...) that way it will obtain proportionality between both regions. Because there are two unknowns (r,n) and two equations (3.1), (3.2). That way, it is possible to adjust the distances and make a consistent proportion between elements.

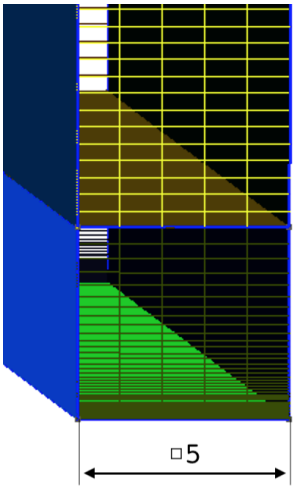


Figure 3.3: The 6 layers of the depth (Y axis).

Taking both equations, it is calculated the number of nodes of every section and the ratio of every mesh. You can found the results in the following table (3.2) . Notice that the result has to be multiplied by 6 because in the depth (Y axis), a separation of 6 layers has been done. As the 5 meters has been divided into 5 sections, what will be obtained is 6 layers in total. Notice that all the values below are an approximation, since some values have been rounded or truncated, like the number of elements or the progression number:

	Regular Separation	Progressive Column			Regular Column			Regular Row			Total Elements
	$dz(m)$	$h_1(m)$	r_1	n_1	$h_2(m)$	r_2	n_2	$h_3(m)$	r_3	n_3	$6\cdot[(n_1+n_2)\cdot n_3]$
Mesh 1	0.15	5	1.03	169	45	1.00	313	200	1.00	250	$\approx$ 723000
Mesh 2	0.20	5	1.04	135	45	1.00	234	200	1.00	250	$\approx$ 553500
Mesh 3	0.25	5	1.055	104	45	1.00	181	200	1.00	250	$\approx$ 427500
Mesh 4	0.30	5	1.07	86	45	1.00	143	200	1.00	250	$\approx$ 343500
Mesh 5	0.40	5	1.085	74	45	1.00	116	200	1.00	250	$\approx$ 285000

Table 3.2: Number of nodes of every mesh proposed.

It is important to know that imposing a fix value of the first separation node ($\alpha \approx 1mm$), means that the y^+ value will remain constant unless the roughness is changed. Because if the roughness changes, the friction velocity too and the y^+ value will be modified. Remembering the friction velocity that Richard and Hoxey propose (1.15) ($U^*_{z_0=1cm} = 0.0481m/s$; $U^*_{z_0=10cm} = 0.0660m/s$; $U^*_{z_0=100cm} = 0.1043m/s$) and the y^+ equation (2.9), the following data is attached:

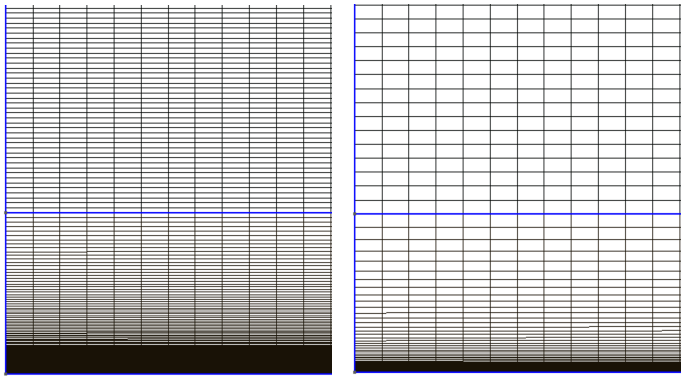
Mesh	$y^+_{z_0=1cm}$	$y^+_{z_0=10cm}$	$y^+_{z_0=100cm}$
1	≈ 3.42	≈ 4.69	≈ 7.41
2	≈ 3.41	≈ 4.68	≈ 7.40
3	≈ 3.29	≈ 4.51	≈ 7.14
4	≈ 3.13	≈ 4.30	≈ 7.01
5	≈ 3.39	≈ 4.65	≈ 7.35

Table 3.3: y^+ values depending on the friction velocity and on the mesh specifications.

A small decimal difference is witnessed between the y^+ value of each mesh since the chosen values of n and r are not 100% perfect. But the difference is practically negligible and it can be considered that they remain in the same y^+ value.

On the other hand, it is seen a displacement of the value as it increases the roughness. This means that, for this mesh conditions, the more roughness is applied, the more abrupt will be the appearance of the buffer layer. As it has indicated in the results chapter, on the figure [5.3].

Meshing with GMSH it can be obtained the five different meshes. Here it can be seen two mesh examples:



(a) $dz = 0.15m$ (X axis) (b) $dz = 0.40m$ (X axis)

Given that figure (a) presents more quantity of number of nodes than figure (b), the calculation time will be greater in (a) than in (b). And although it seems that the figure (a) will ensure more precision and accuracy in the results than (b), you can not be sure of that until they are put to the test in the same simulation with the same boundary conditions. Remember that a finer mesh guarantees good results but it is not optimal in the simulations, since it is an unnecessary expense of calculation time.

Once the meshes have been created, it is important to verify them and to check them in the OpenFOAM environment. In the terminal you order it to read the file by writing ”gmshToFoam mesh01.msh”, that is the file that contains the details of the mesh with $dz = 0.15m$ of separation between nodes in the regular zone. Once read the file, you have to verify if the mesh is correct by the program by writing ”checkMesh”. And it is obtained the following information:

```
Checking geometry...
    Overall domain bounding box (0 0 0) (200 5 50)
    Mesh has 2 geometric (non-empty/wedge) directions (1 0 1)
    Mesh has 2 solution (non-empty) directions (1 0 1)
    All edges aligned with or perpendicular to non-empty directions.
    Boundary openness (2.90138283769e-17 9.2888659727e-19 -2.53900604245e-16) OK.
    Max cell openness = 2.20897944637e-16 OK.
    Max aspect ratio = 949.051146407 OK.
    Minimum face area = 0.000679524689199. Maximum face area = 0.803212897723.  Face area magnitudes OK.
    Min volume = 0.000679524689199. Max volume = 0.155816398643.  Total volume = 49999.9999999.  Cell volumes OK.
    Mesh non-orthogonality Max: 0.0338996641444 average: 0.00615051169376
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 0.000848897549316 OK.
    Coupled point location match (average 0) OK.

Mesh OK.

End
```

Figure 3.4: OpenFOAM mesh verification.

It is observed that OpenFOAM identifies well the geometry and its mesh, so now it would be necessary to propose some fixed boundary conditions and go choosing the different meshes to see their differences in the results.

3.1.2. Mesh Validation

Proceed then, to the simulations with the boundary conditions described above. Since the initial velocity will always be the same, the only thing that can intervene in the modification of the results, is the roughness. By modifying the roughness, you modify the turbulence and that means that the boundary conditions of an initial simulation are not the same as another simulation with different roughness. Therefore, three simulations will be carried out for the convergence of the mesh. One with roughness of 1cm, another of 10cm and 100cm. These three values have been chosen to put the meshes in limit conditions, that is, both in conditions of very low turbulence and very high turbulence.

Once the boundary conditions and the initial conditions of the simulation have been specified, the iterations are calculated. The data resulting from the speed are extracted at the outlet of the computational domain for each mesh. That is, the results at $x = 200m$ length. After a few hours (≈ 3000 iterations) it is got the following:

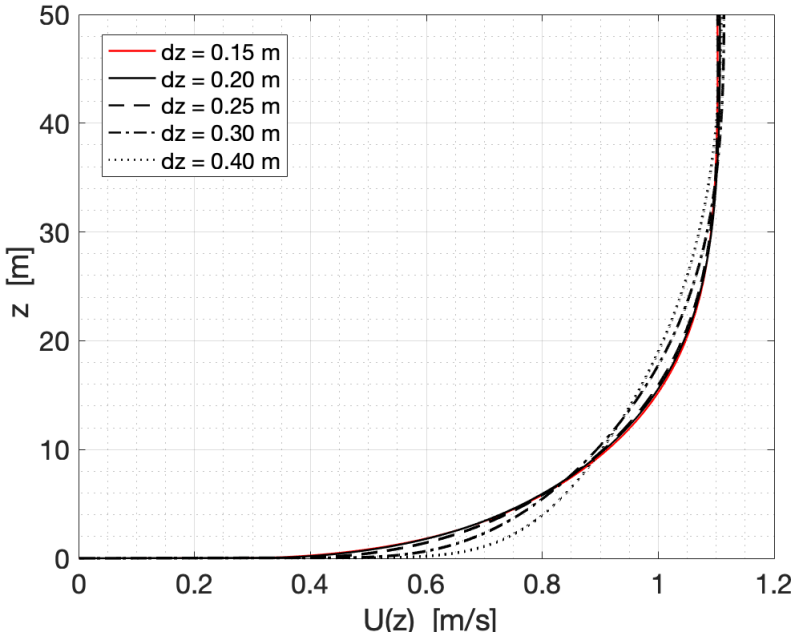


Figure 3.5: Outlets at $x = 200m$ with $z_0 = 10cm$

It can be seen that the resulting velocity profile adopts a convergent shape as the mesh increases the number of nodes. At first glance it can be said that the velocity profile of the finest mesh, represented in red, does not present many differences with respect to its previous mesh. So in these cases it is intuited that the mesh selected to make future simulations will be that of $dz = 0.20m$. But before that, a relative analysis must be made between the solutions to reach a conclusion. Because these are for roughness of 10cm and they can be different for the other roughnesses.

Notice that the results of the $z_0 = 1cm$ and $z_0 = 100cm$ are attached in the [Annex A] and the Figure [3.5] is an example.

3.1.3. Convergence Error Analysis

For the convergence error, it shall be chosen the mesh that is able to compute the results well without needing to apply more calculation points. One way to do this is to calculate the relative error that each mesh commits with respect to the finest one and choose the one that has less error. As in this case it is working with vectors, they should be normalised in order to obtain the relative error.

The scalar relative error is described like:

$$\epsilon = 100 \cdot \frac{|x_{approx} - x_{real}|}{x_{real}} \ [\%] \tag{3.3}$$

Where x is a scalar value in this case. Now if it works with vectors, the expression of the relative error is:

$$\epsilon = 100 \cdot \frac{||(\vec{x}_{approx} - \vec{x}_{real})||}{||\vec{x}_{real}||} \ [\%] \tag{3.4}$$

Where ϵ is a scalar and $||\vec{x}||$ is the vector norm of x in this case. Since what is wanted is to compare the finest mesh with the other ones, it is set:

$$\vec{x}_{real} = \vec{M}_{dz=0.15\,(m)}$$
$$\vec{x}_{approx} = \vec{M}_{dz=K\,(m)}$$

Where $K = [0.20, 0.25, 0.30, 0.40]$. There are many ways to normalise a vector, but the best known are the following:

- L^1 vector norm

$$||x||_1 = \sum_{i=1}^{n} |x_i| \tag{3.5}$$

- L^2 Euclidean vector norm

$$||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{3.6}$$

- L^∞ vector norm

$$||x||_\infty = max\ |x_i| \quad i = 1, \ldots, n \tag{3.7}$$

Using equation (3.4) it is obtained the following results for the three types of normalization:

dz(m)	$\epsilon(\%)\,by\,L^1$	$\epsilon(\%)\,by\,L^2$	$\epsilon(\%)\,by\,L^\infty$
0.20	**0.204**	**0.230**	**1.562**
0.25	0.948	1.391	6.096
0.30	1.907	2.924	12.813
0.40	2.602	3.910	15.985

Table 3.4: Error for different vector norms of $z_0 = 1cm$.

dz(m)	$\epsilon(\%)\,by\,L^1$	$\epsilon(\%)\,by\,L^2$	$\epsilon(\%)\,by\,L^\infty$
0.20	**0.305**	**0.251**	**1.366**
0.25	0.525	1.026	5.934
0.30	1.892	2.861	13.194
0.40	2.920	4.621	19.410

Table 3.5: Error for different vector norms of $z_0 = 10cm$.

dz(m)	$\epsilon(\%)\, by\, L^1$	$\epsilon(\%)\, by\, L^2$	$\epsilon(\%)\, by\, L^\infty$
0.20	**0.096**	**0.010**	**0.401**
0.25	1.329	1.738	8.772
0.30	2.603	3.719	17.192
0.40	3.710	5.644	24.199

Table 3.6: Error for different vector norms of $z_0 = 100cm$.

As it can be seen, the more nodes, the less error. Therefore, the mesh that best fits the finest mesh ($dz = 0.15m$), is the mesh that has an amount of 553500 nodes and a separation of $dz = 0.20m$ in the regular area. Notice that this mesh has a first cell distance value of $0.97\,mm$ so it gets a y^+ value of $y^+ \approx 1$ approx.

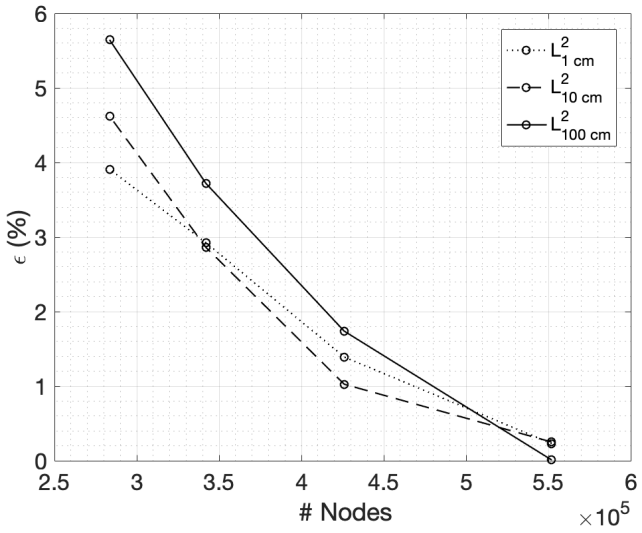


Figure 3.6: Relative error vs number of nodes of the mesh, using L^2 norm.

It can be concluded then that the mesh with the best calculation time ratio per number of nodes is that of $dz = 0.20m$. Therefore, from now on, the characteristics of this mesh will be respected regardless of the geometrical modifications that may be created.

3.2. Boundary Conditions

As explained in the theoretical part [1.4.], Richard and Hoxey proposed two types of boundary conditions for the input of the computational domain: Uniform and Non-Uniform inlet conditions. One consists in introducing the theoretical logarithmic profile directly (ABL equation) and the other consists in fixing a constant value of velocity in the whole surface.

According to the boundary conditions that OpenFOAM provides, for each surface and for each physical quantity, the possible layers are attached below. Notice that the underlined ones are those that must be confirmed for final determination.

Boundaries	U_x	P
Inlet	fixedValue or atmBoundaryLayerInletVelocity	zeroGradient
Top	fixedValue or slip	slip
Outlet	inletOutlet	uniformFixedValue
Bottom	uniformFixedValue	zeroGradient
Front & Back	symmetry	symmetry

Table 3.7: Velocity and pressure boundary conditions.

Boundaries	k	ϵ	ν_t
Inlet	atmBoundaryLayerInletK	atmBoundaryLayerInletEpsilon	calculated
Top	slip	slip	slip
Outlet	inletOutlet	inletOutlet	calculated
Bottom	kqRWallFunction	epsilonWallFunction	nutkAtmRoughWallFunction
Front & Back	symmetry	symmetry	symmetry

Table 3.8: Turbulence and roughness boundary conditions.

The first simulations will consist, then, in observing the differences presented by the different boundary conditions underlined. There are four different simulations that are compared with the theoretical model (1.12) in order to see which of them present less error:

- Uniform Inlet (fixedValue) & Uniform Top (fixedValue)
- Uniform Inlet (fixedValue) & Non-uniform Top (Slip)
- ABL Inlet (atmBoundaryLayerInletVelocity) & Uniform Top (fixedValue)
- ABL Inlet (atmBoundaryLayerInletVelocity) & Non-uniform Top (Slip)

3.2.1. Selection of the inlet and top boundary conditions.

To know which one choose from the four proposed options, the output velocity profile will be compared with the theoretical model or the ABL equation. The study will continue with those options that present less error.

As in this section what is studied are the boundary conditions, it is not crucial yet the selected roughness. In this case, to have a reference value, the following simulations will be performed with $z_0 = 10cm$ roughness. The selection of the roughness will be found in the results [CHAPTER 5].

In the first case, a uniform inlet is introduced along the z-axis. And two simulations will be made with this inlet. One with a constant top and the other with a slip top.

Keep in mind that if a constant and uniform inlet velocity is introduced over the entire height, the kinetic energy will not be the same as that of a logarithmic profile. So it must be applied the principle of conservation of the mass so as to respect this energy and thus be able to adapt the logarithmic profile ABL to the uniform inlet. Following

the example attached in the [Appendix B], it is possible to get the ABL equation in the figure below.

The corresponding data is written in the OpenFOAM terminal and the simulation starts and continues until it reaches the convergence. Below the following results:

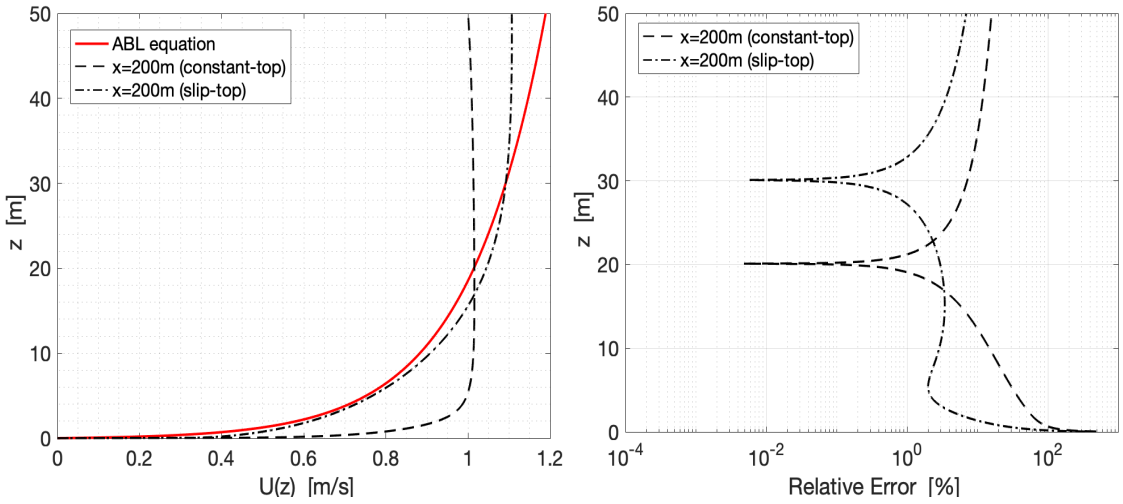


Figure 3.7: Uniform inlet with two different tops: constant and slip

By means of the analytical calculation of the relative error that each outlet makes with respect to the ABL equation, two peaks can be observed that tend to an error of 0.01%. These peaks would have to tend to 0% error, since it is when the two discontinuous graphs intersect with the theoretical model. This is not the case since the data extracted from Paraview does not include all the corresponding decimals.

On the other hand, the plot that commits the least relative error is that of the slip-top case. This is due to the fact that by placing a sliding roof, it makes the fluid more easily adopt a logarithmic shape and in this case, more like to the theoretical one. On the other hand, when a top-constant is imposed, its freedom is limited and it's practically the entry profile.

Finally, in both cases, at the bottom it can be seen a very high error. This may be due to the transition between the laminar and the turbulent velocity profile. That is, the passage through the "Buffer Layer".

In the previous simulation it was possible to see that once the speed converged, neither of the two cases could adopt 100% the logarithm in question. What it is wanted to check now, is if OpenFOAM is able to respect the theoretical model of Richard and Hoxey during the course of the simulation. The inlet boundary conditions are simply changed and the ABL function is introduced instead of the constant speed. Since in this case the input has been the equation ABL, it is not necessary to adapt it to any different kinetic energy, because it is the same for all cases. So the following results are obtained:

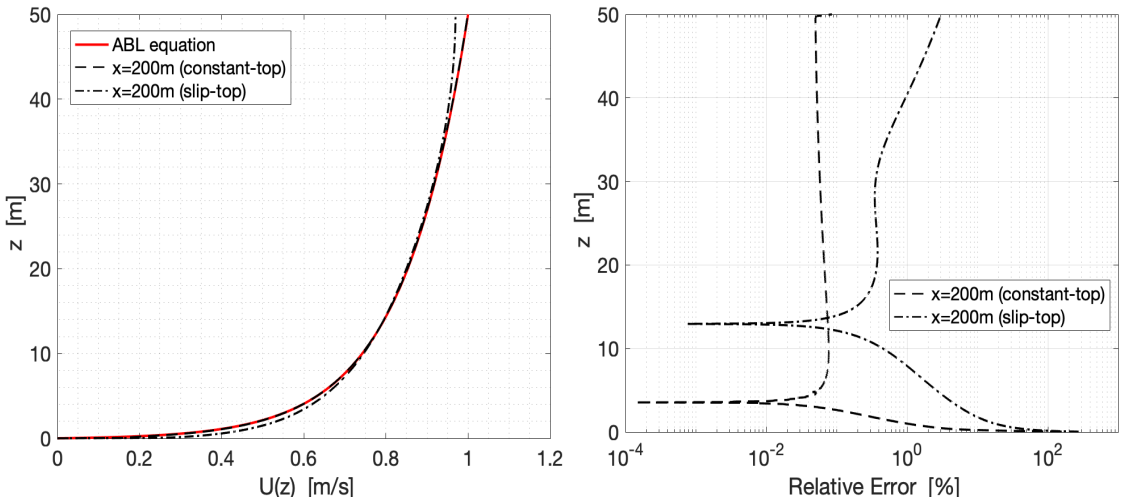


Figure 3.8: ABL inlet with two different tops: constant and slip

In this case it can be seen that the difference between the simulations and the theoretical model are minimal. Something better is that of the top-constant condition since you ensure that the logarithm is set as the maximum velocity that shows the equation ABL. Anyway, the maximum error that is obtained, filtering the errors that are shown in the lower part, is a maximum of 4%.

If you compare the relative errors between the uniform inlet and the ABL inlet, it can be concluded that regardless of the condition you impose on the top, they will always give you better results in the second case:

Cases	Relative Error Interval (%)
Uniform inlet & Constant-top	$[5,30]$
Uniform inlet & Slip-top	$[1,10]$
ABL inlet & Constant-top	$[0.01,0.1]$
ABL inlet & Slip-top	$[0.1,5]$

Table 3.9: Boundary conditions relative error.

To sum up, it can be said that the boundary conditions for the top cover, good results are obtained regardless if it is slip or constant. On the other hand, a uniform inlet is not efficient for the initialisation of a simulation, since the output does not represent an analytical logarithmic.

That is why the final selection of the boundary conditions for all simulation would be the following:

- ABL inlet & Constant-top
- ABL inlet & Slip-top

The peak of error that is observed in the plot is caused by the Buffer Layer (the region will be demostrated in the roughness study [5.3]) and it can be see that it happens at a height of approximately $50cm$. Since we would never physically place a wind turbine so small, this peak of error could be filtered and it won't affect to the next simulations.

CHAPTER 4. SOLVER

This chapter explains the strategy that has been chosen within the OpenFOAM program. Since this program has many solvers in its database, it is important to know which one to choose depending on the physical properties of the fluid to be studied. It is also important to know how it works and what is its algorithm to use.

4.1. SimpleFoam Solver

4.1.1. General properties

SimpleFoam is a steady-state solver for incompressible, turbulent flow, using the SIMPLE algorithm in OpenFOAM. The solver follows a segregated solution strategy. This means that the equations for each variable characterising the system (the velocity $\vec{U}$, the pressure P and the variables characterising turbulence) is solved sequentially and the solution of the preceding equations is inserted in the subsequent equation.

SimpleFoam is defined by OpenFOAM [23] as: *Steady-state solver for incompressible, turbulent flow*. And it can be used for the following characteristics of any fluid:

- Incompressible
- Steady-State
- Laminar and Turbulent
- Single phase
- Isothermal

So the equations used are (1.3) and (1.5):

$$\begin{aligned} \nabla \cdot u &= 0 \\ \nabla \cdot (uu) &= -\frac{1}{\rho}\nabla \vec{p} + \nabla \cdot \nu + \vec{F_m} \end{aligned} \tag{4.1}$$

4.1.2. Explicit and implicit methods

There are many ways to calculate states from others already calculated. But as the simple algorithm uses the explicit method and the implicit method.

The explicit method allows to calculate the next value of any physical variable depending on the current value. That is, it allows calculating the later state of the system from its current state. Analytically, if $Y(t)$ is the current system state and $Y(t+\Delta t)$ is the state at the later time (Δt is a small time step), then, it can be relate them in the following way:

$$Y(t+\Delta t) = F(Y(t)) \tag{4.2}$$

On the other hand, the implicit method to solve the posterior value will need to solve an equation that takes into account the current state and the previous state of the system. So the equation will consider another function:

$$Y(t+\Delta t) = F(Y(t+\Delta t)) + G(Y(t)) \tag{4.3}$$

Using the implicit method will suppose to make an extra calculation and therefore a delay in the simulation.

4.1.3. SIMPLE algorithm

SimpleFoam uses the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm.

This algorithm takes into account a sequence of calculation that allows to stabilise and specify the data of the problem looking for a convergent solution.The main functions of this algorithm are the following:

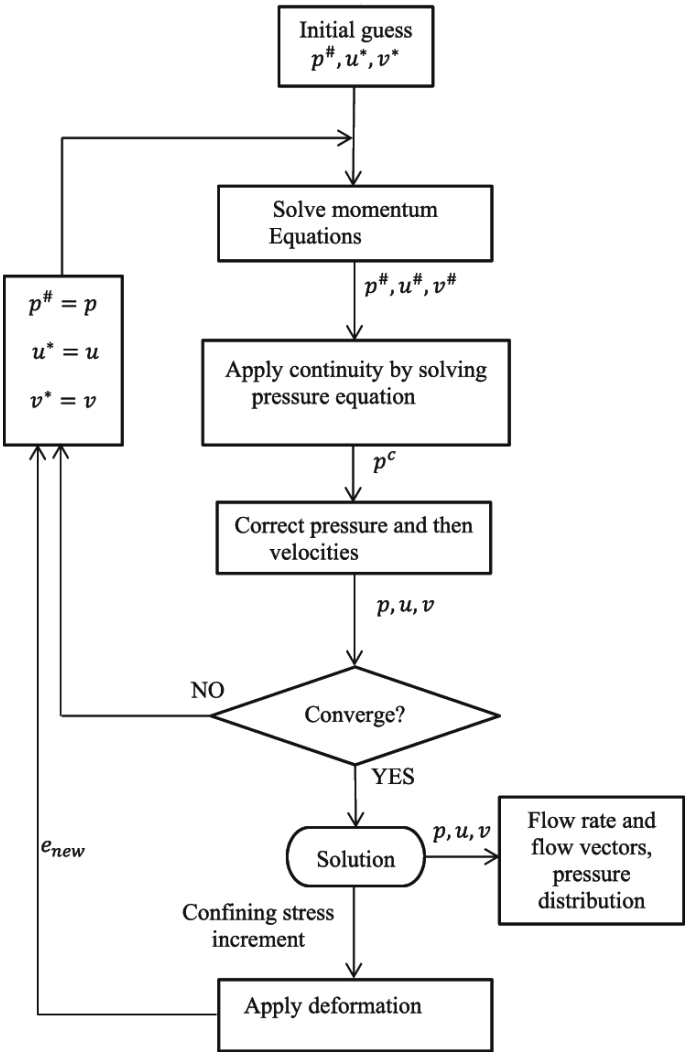


Figure 4.1: Calculation process of SIMPLE algorithm. [24]

- First the momentum equation is solved, in order to obtain an approximation in the velocity. The pressure gradient is calculated using the pressure value of the initial or previous iteration.
- So the new pressure is calculated in order to have a distribution in the current iteration.
- Velocities are corrected and then it can be obtained the conservatives fluxes.

This algorithm has the name of a ”Semi-Implicit Method” because the discretisation of the momentum equation and the pressure one, are solved implicitly. And the velocity correction is solved explicitly. So a mix of methods are found.

CHAPTER 5. RESULTS

In this chapter it is found the solutions of the simulations that are carried out, where the previous theoretical concepts are put into practice, as well as the use of the convergent mesh. The chapter is developed by means of logical answers to different questions that arise to the obtained results. First, the influence of roughness is known and then it continues simulating with other geometries in the computational domain. In each section of the chapter a different study is developed obtaining a result that is decisive for the progress of the work.

5.1. The influence of roughness

The intention of this study is to see the influence of roughness on the velocity profile. Taking as reference the theoretical section [1.1.1.], it is decided to choose three different roughness values: $z_0 = 1cm$, $z_0 = 10cm$, $z_0 = 100cm$. Two extreme values and one intermediate in order to see their differences.

For the simulation, the same geometry that was used for the validation of the mesh will be used. A length of 200 meters and a height of 50 meters, respecting the mesh values selected in the previous chapter.

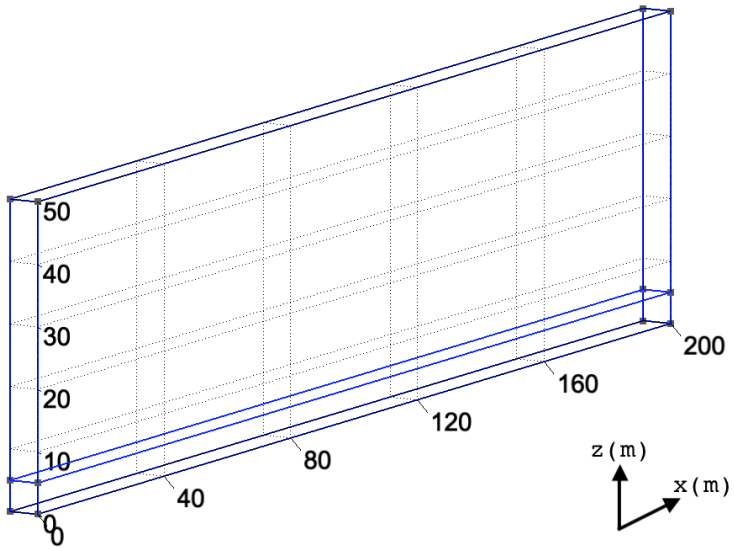


Figure 5.1: Basic geometry.

The roughness length of $z_0 = 1cm$ represents a slightly rough terrain, which has very little friction force. On the other hand, the value of $z_0 = 100cm$ represents a surface with urbanisation, which it will cause the fluid to be considerably affected. The intermediate value of $z_0 = 10cm$ of roughness would be the one that would represent farm land with bushes and trees. Since the work is focused on research in wind energy, this value would be the one that would most closely approach the proposed land.

Keep in mind that by changing the roughness, the initial k and ϵ values will change. A 2007 study by the *American Institute of Aeronautics and Astronautics* [21] , says that in the face of an external fluid and in this case the atmosphere behaviour, the eddy viscosity ratio μ_t/μ has to be between 0 and 10. Usually get a value close to 5. And this is the value that it has been chosen for the simulations. So, by the relationship between the turbulence and the roughness of equation (2.5) and imposing that $\mu_t/\mu = 5$ it can be seen:

$$\frac{\mu_t}{\mu} = 5 \rightarrow \mu_t = 1.802 \cdot 10^{-5} \cdot 5 \approx 9 \cdot 10^{-5} kg/ms \tag{5.1}$$

$$\mu_t = \rho C_\mu \frac{k^2}{\epsilon} \rightarrow \epsilon = \frac{C_\mu k^2 \rho}{\mu_t} \tag{5.2}$$

In this way, it can be calculated the initial value of epsilon respecting the eddy viscosity ratio proposed using the equation (5.2). And for the calculation of the k, it is used the equation (1.13) specified in the theoretical chapter. Therefore according to this, it can be obtained the following turbulence values that will be used:

$$z_0 = 1cm \rightarrow k = 0.0074 \ \& \ \epsilon = 0.0662$$
$$z_0 = 10cm \rightarrow k = 0.0138 \ \& \ \epsilon = 0.2333$$
$$z_0 = 100cm \rightarrow k = 0.0345 \ \& \ \epsilon = 1.4580$$

The boundary conditions are those that have been concluded in the previous [CHAPTER 3], in the [3.2.1.] section. The initial turbulence values will be those calculated previously and the input velocity will be the same conditions for all of them, imposing that $U(Z_{ref} = 50m) = 1m/s$, as in the rest of the simulations (justification explained in the inflow conditions [1.4.1.]). For the rest of heights, the value of the speed will vary depending on k and ϵ.

In addition to seeing the influence of the roughness in the profile, the speed resulting from the simulation will be compared with the theoretical velocity ABL. In this way it can be checked if the simulations are able to correctly approach the profile that Richard and Hoxey managed to model.

5.1.1. Top-constant case

For this section of the study, a top cover with a constant speed along the entire route is imposed in the domain. The fixed values are: $z_{top} = 50m$ with velocity $U(z_{top}) = 1m/s$. Notice that the inlet is the ABL velocity log-profile as it was concluded in the section [3.2.1.].

So the resulting profiles that are obtained are the following:

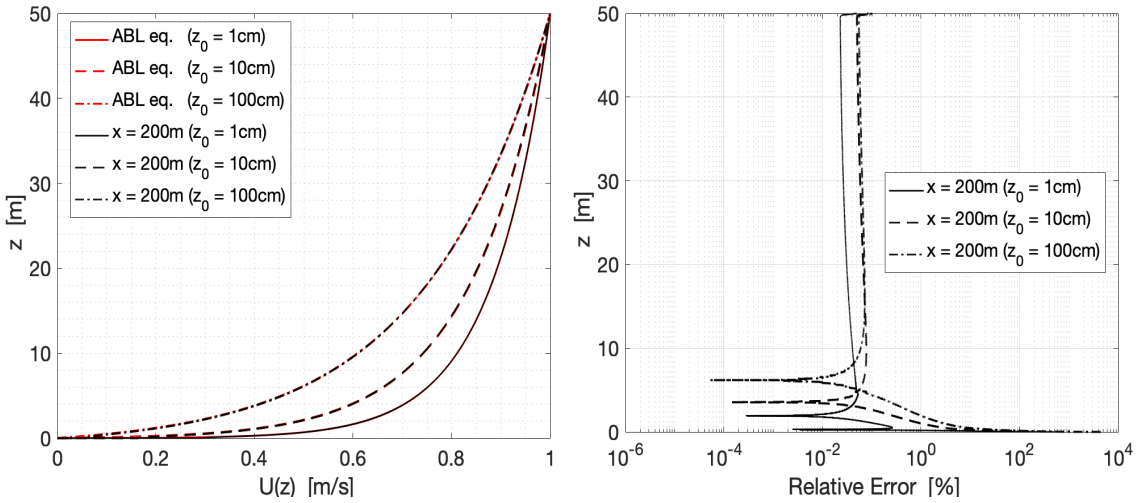


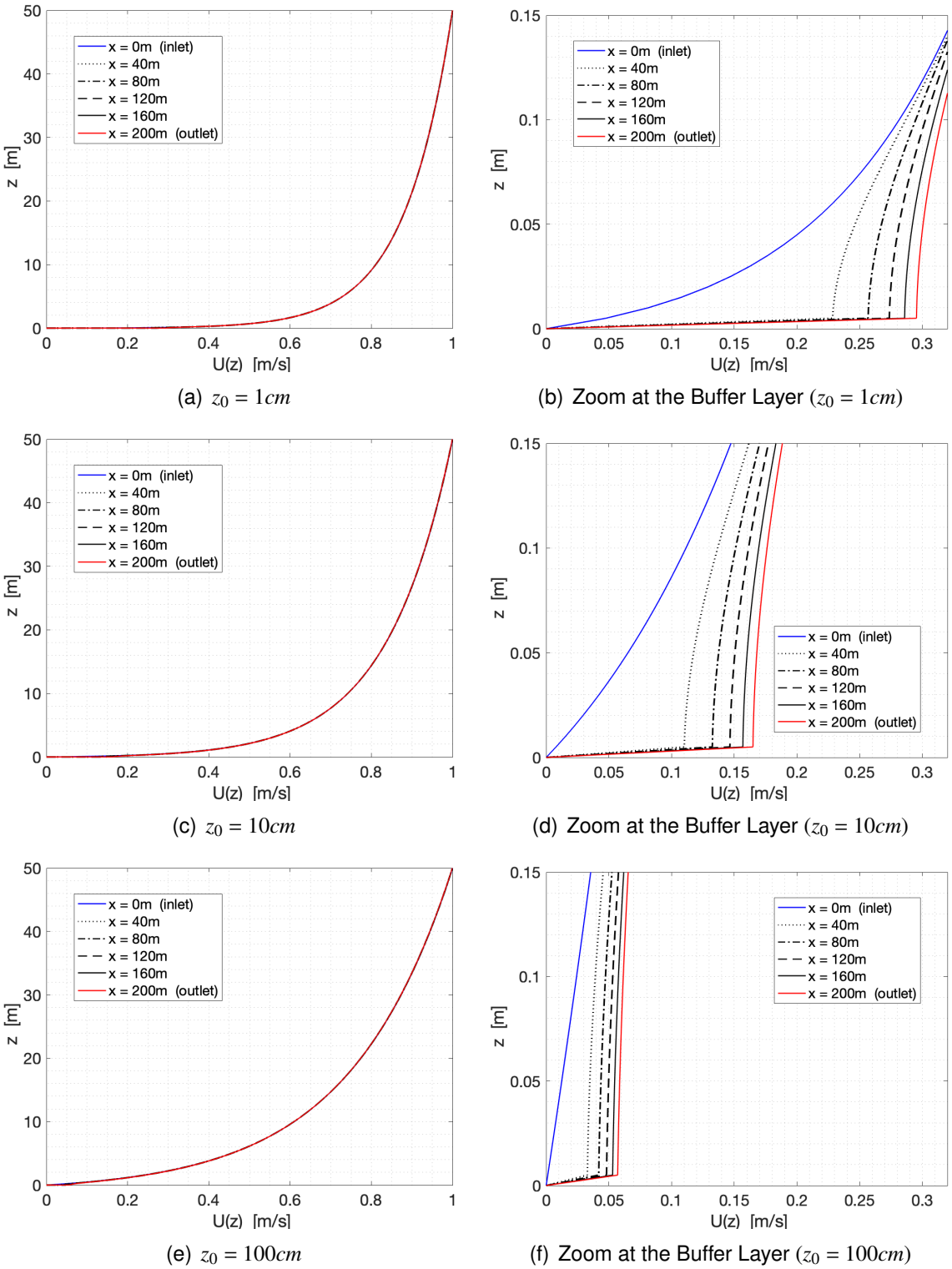
Figure 5.2: Differences between the outlets of different roughness with the ABL equation. (Top-constant)

According to the outlet speeds obtained, it can be seen that the more roughness, the less speed it presents. This is due to the hindrance represented by the roughness, which makes the friction greater.

The three profiles respect the condition of $1m/s$ at 50 meters of height, so the top is constant for all three cases, as expected. This means that in the rest of the heights, the differences in speeds are easily shown.

From the plot of the relative error it can be concluded that the more roughness, the more error is obtained in the approximation of the velocity profile. For roughness $z_0 = 1cm$, the average relative error is between 0.10% and 0.20%. On the other hand, for $z_0 = 10cm$ and $z_0 = 100cm$, both are between 0.40% and 0.50%, taking into account that for $z_0 = 10cm$ is slightly lower. Regardless of their differences, it can be concluded that the error is practically negligible with respect to the theoretical profile.

At the bottom it can be seen an abrupt peak of error. This could be because of the transition of the profile, when it exceeds the Viscous Sublayer to the Buffer Layer. To ensure that this is true, the evolution of the velocity along the X axis is shown below, expanding the lower area of the plot:



(a) $z_0 = 1cm$

(b) Zoom at the Buffer Layer ($z_0 = 1cm$)

(c) $z_0 = 10cm$

(d) Zoom at the Buffer Layer ($z_0 = 10cm$)

(e) $z_0 = 100cm$

(f) Zoom at the Buffer Layer ($z_0 = 100cm$)

Figure 5.3: Buffer Layer of different roughness along the x-axis. (Top-constant)

In order to check whether these regions really belong to the Buffer Layer, it must be verified that the dimensional values of $z(m)$ are within the range of the non-dimensional values of z^+ specified in the section [2.2.1.]. In order to get a non-dimensional height ($z^+ \equiv y^+$ at a conceptual level), it is necessary to apply the equation (2.9).

For every roughness there is a different friction velocity so the z^+ value will be different.

Looking closely at the zoom figures, it can be seen that the maximum no-fitting value, is shown at the height of $z \approx 0.005m$. Therefore, if the values of z^+ at the height of $z = 0.005m$ is within the range of $z^+ = [5, 30]$, then the region is in the Buffer Layer. Using equation (1.15) for every roughness ($z_0 = 1cm$; $z_0 = 10cm$; $z_0 = 1cm$), it can be obtain the following data:

$z_0(m)$	$U^\star$ (1.15)	$z^+ = U^\star \cdot 0.005/\nu$
0.01	0.0481	≈ 16.36
0.1	0.0660	≈ 22.44
1	0.1043	≈ 35.47

Table 5.1: z^+ values inside the Buffer Layer

The z^+ values for each roughness at the height of $z = 0.005m$ are within the range of the Buffer Layer. If the plots are compared with the z^+ values, it can be seen that the higher the roughness, the further away from the Buffer Layer. The value of $z^+ = 35$ is on the border and in the plots it can be checked that the $z_0 = 100cm$ fitting, is better than $z_0 = 1cm$ one.

Anyway, this error could be filtered without any type of problems, since the wrong fitting is between $z = 0m$ and $z = 0.10m$ high, and in this area is impossible to place a wind turbine and it would be physically unfeasible. So the study can be continued ignoring the error that produces the Buffer Layer.

5.1.2. Top-slip case

In this case, the initial conditions are exactly the same as for the previous case but changing the top boundary condition to the slip one. The results obtained are below:

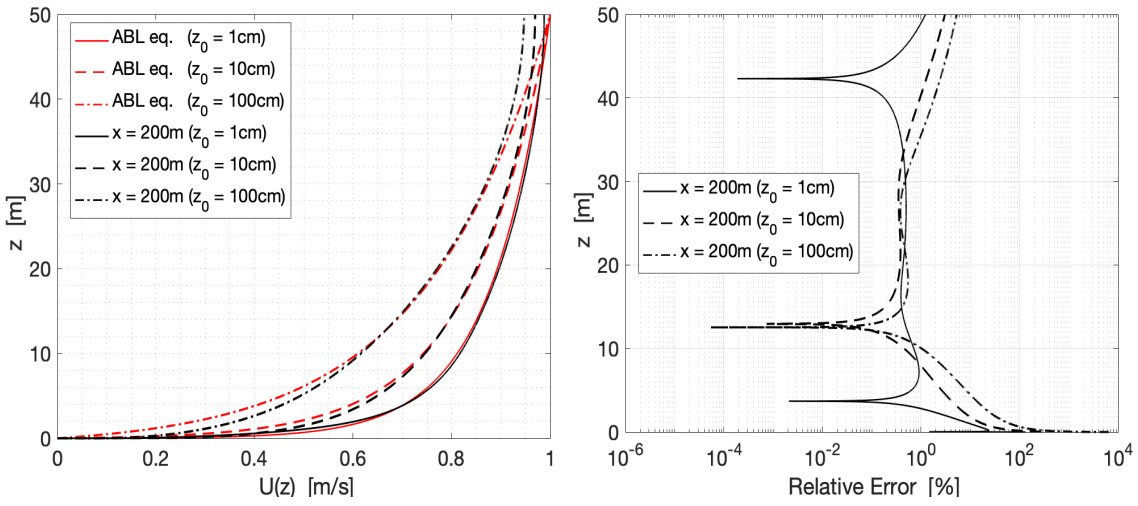


Figure 5.4: Comparison between the outlets of different roughness with the ABL equation. (Top-slip)

Observing the plot of the relative error it can be said that in this case a difference appears with respect to the previous case. The error is very similar for the three different roughness between the $z = 15m$ and $z = 35m$ height. And in the rest of heights it can be seen that the more roughness, the more error presents as well as in the top-constant case.

To sum up it is concluded that the greater the roughness, the more turbulence is obtained and this makes the error greater. Also, the constant top is still better than the sliding top although its approach is still good.

Since it is physically impossible to place a wind turbine at a height of less than 1 meter, the error produced by the Buffer Layer is filtered without problems. The following studies will continue respecting the same boundary conditions until new geometries can cause inconvenient for the results.

As for the value of the roughness, the length of 10cm is chosen for the rest of the simulations, since the intention of the work is to observe the flow of wind through mountains and valleys. And this roughness is the one that best approximates the conditions of the terrain.

5.2. Ascending Slope

In the following study, what is proposed is to visualise the behaviour of the velocity profile when a slope is found. Two slopes are proposals of 10% and 20%. The path is 200 meters length and 50 meters height, when applying a slope of 20%, the height at the exit is severely impaired and could modify the behaviour of the fluid:

$$slope = 0.20 = \frac{\Delta z}{\Delta x}$$
$$\Delta z = 0.20 \cdot 200 = 40m$$

Let's see the resulting data when a 20% slope is applied. Notice that the slip and constant conditions are kept because it is desired to see the difference between slope influence and flatness influence:

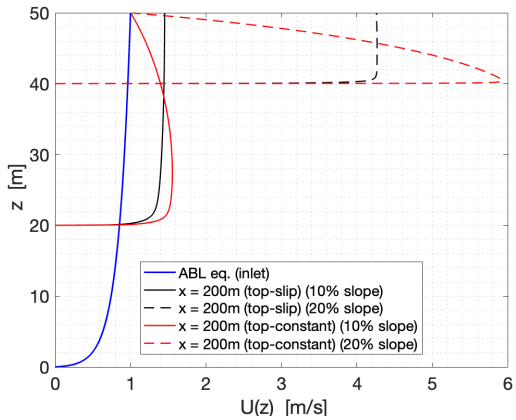
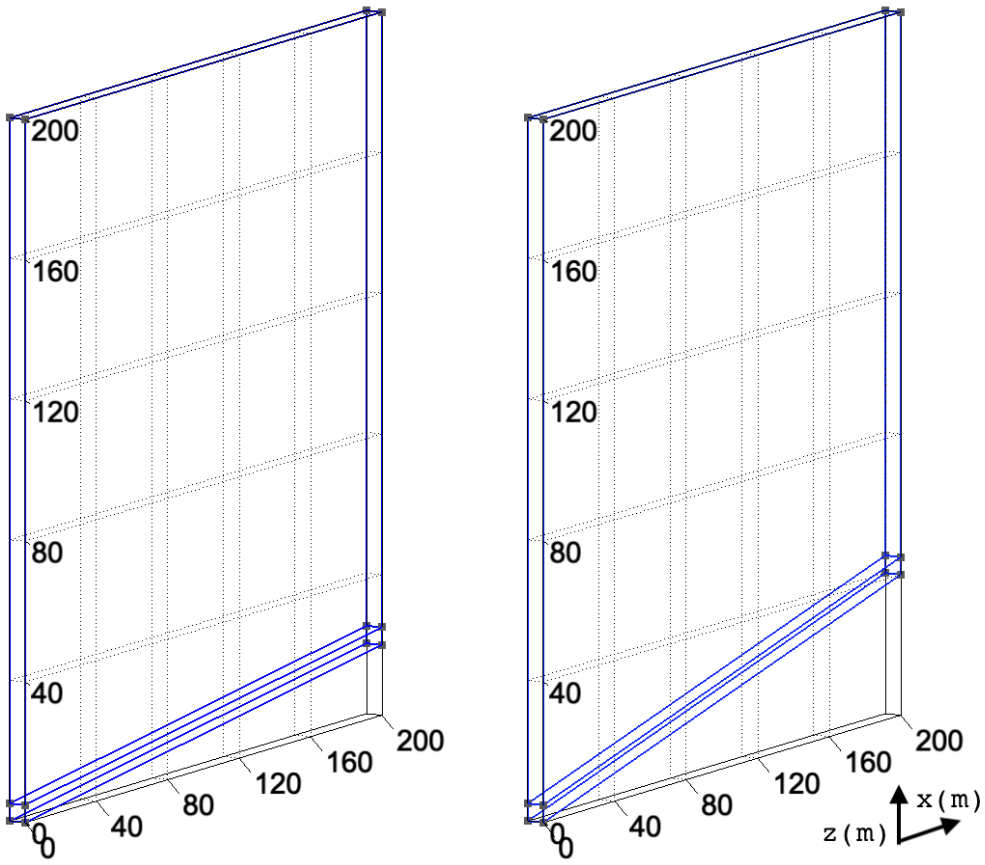


Figure 5.5: Outlets of different slopes with different top boundary conditions. ($z_{top} = 50m$)

It can be seen that if a slope of 20% is applied maintaining the height of $z_{top} = 50m$, the fluid behaves as if it were passing through a funnel since the top layer is behaving like a wall. So the outlet area is reduced and the velocity is higher.

That's why it has been increased the height of the computational domain in $z_{top} = 100m$, $z_{top} = 200m$, $z_{top} = 300m$ and $z_{top} = 400m$, because the objective is to know the impact of increasing the height of the domain and avoid this irregularities.

(*Similar conditions that the Budapest University of Technology and Economics researchers did on January 4th, 2017 when was the submission of the project proposal [4]. See Figure 1.3.*)



(a) $Slope = 10\%$ (b) $Slope = 20\%$

Figure 5.6: Slopes implementation by GMSH ($z_{top} = 200m$)

5.2.1. Mesh regulation

The geometry will be affected by a slope and by different height increments. That's why it must be borne in mind that the characteristics of the mesh selected in the study of the convergence, must be respected and adapted in these new conditions. If the height of the top boundary increases, the top separates more from the turbulent zone and the mesh can be generated with fewer node points in that region, since it is not affected as much as before. So now there would no longer be a regular zone but a progression with increased separation between nodes as it approaches the top.

Therefore, with respect to mesh $dz = 20m$ everything remains the same (135 elements for the 5 first meters tall and 250 elements for the 200 meters length) except the following data for the 10% of slope case:

	Inlet and Outlet Column			Total Elements
	$h_{inlet}(m)$ / $h_{outlet}(m)$	r_{inlet} / r_{outlet}	n	$6 \cdot [(135+n) \cdot 250]$
$z_{top} = 100m$	95 / 75	1.0095 / 1.0080	200	$\approx$ 502500
$z_{top} = 200m$	195 / 175	1.0045 / 1.0040	400	$\approx$ 802500
$z_{top} = 300m$	295 / 275	1.0030 / 1.0030	600	$\approx$ 1102500
$z_{top} = 400m$	395 / 375	1.0024 / 1.0023	800	$\approx$ 1402500

Table 5.2: Slope 10%

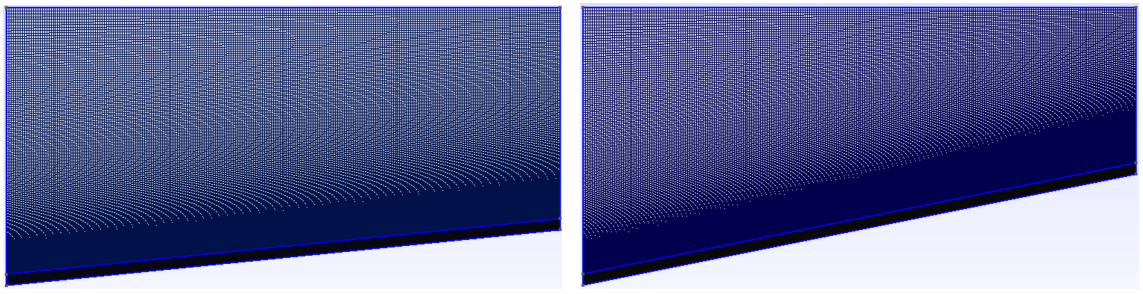
	Inlet and Outlet Column			Total Elements
	$h_{inlet}(m)$ / $h_{outlet}(m)$	r_{inlet} / r_{outlet}	n	$6 \cdot [(135+n) \cdot 250]$
$z_{top} = 100m$	95 / 55	1.0095 / 1.0070	200	$\approx$ 502500
$z_{top} = 200m$	195 / 155	1.0045 / 1.0038	400	$\approx$ 802500
$z_{top} = 300m$	295 / 255	1.0030 / 1.0029	600	$\approx$ 1102500
$z_{top} = 400m$	395 / 355	1.0024 / 1.0021	800	$\approx$ 1402500

Table 5.3: Slope 20%

For the slope of 20% it would be the same but changing the progression in the output, and the number of nodes is exactly the same as in the 10%. Different values of progression because having more slope, the section of the output is lower.

Notice that the 6 layers of the Y axis are still being maintained to ensure that the velocity profile remains the same regardless of the slope. It will not be removed until it is definitively concluded that the change can be neglected.

Here it can be seen an example of the resulting meshes:



(a) Slope of 10% (b) Slope of 20%

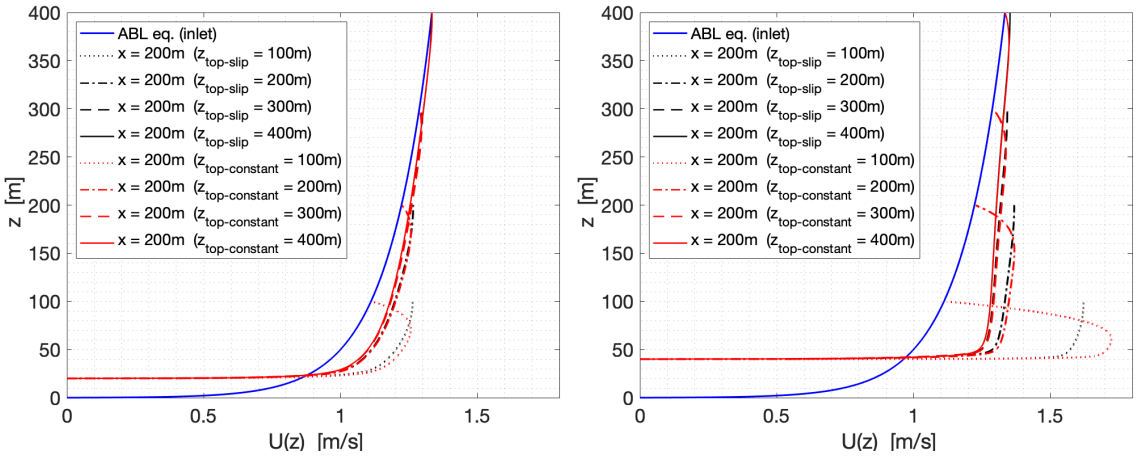
Figure 5.7: Meshes with different slopes ($z_{top} = 100m$)

Respecting then the concentration of nodes of the selected mesh guaranteeing the separation of $dz = 0.20m$ in the approximately first 50m high, OpenFOAM will arrange to calculate the velocity profile with these 4 different meshes for the 4 different top heights in a different percentage of slope.

Regarding the development of the speed in the axis in depth, it can be consulted in the [Annex C], which has no influence and therefore the mesh can be generated in almost 2D. In this annex it can be seen the results obtained along the y-axis, both for the case of the plain, and for the slope.

5.2.2. Outcomes

Once the meshes have been created, the simulations are ready to be performed. The exact same thing is done for the previous case but with the heights of $z_{top} = 100m$, $z_{top} = 200m$, $z_{top} = 300m$, $z_{top} = 400m$ getting the following results:



(a) Slope of 10% (b) Slope of 20%

Figure 5.8: Outlets of different slopes with different top boundary conditions.

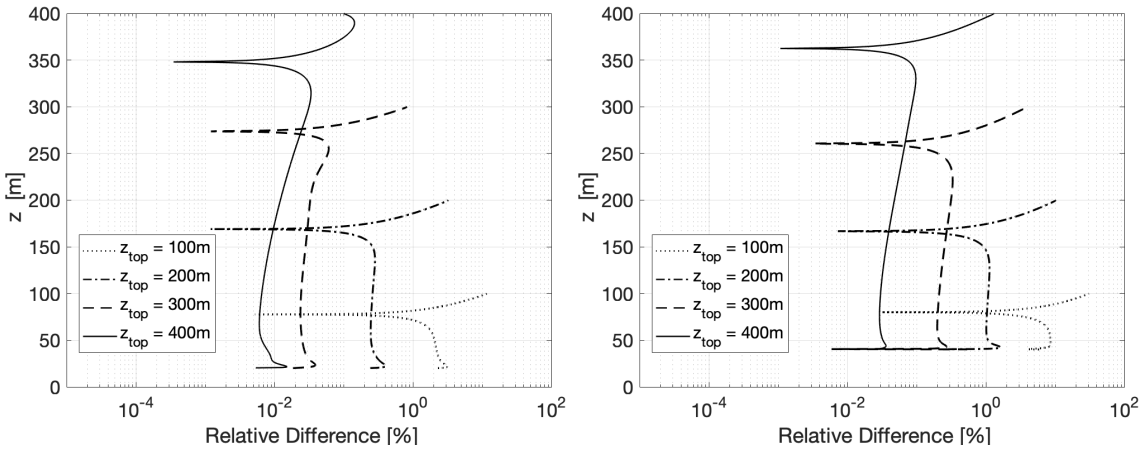
As the height of the top boundary is increased, the funnel effect disappears and more logical results for the interpretation of a slope mountain is obtained. Also it can be found more acceleration when the slope percentage is increased:

The higher the percentage of the slope, the greater the reduction of the cross-sectional area along the x-axis. This reduction of area causes the flow to accelerate due to the principle of conservation of the mass [1.2.2.]. That is to say, in order to transport the same amount of air in the same period of time, from a larger initial area to a smaller final area, the flow must accelerate.

Another detail to take into account, is that as the height increases, the condition of top-slip and top-constant are getting a negligible difference. In order to analyse the differences between them, it is calculated the relative difference that has the constant condition with respect to the sliding condition for each height. Since it is not being compared with a theoretical solution, the difference that is obtained is not an error. And for that reason what is calculated is a *relative difference* between both top boundary conditions:

$$Dif(\%) = 100 \cdot \frac{abs(U_{top-slip} - U_{top-constant})}{U_{top-slip}} \tag{5.3}$$

The calculation is applied for each height and the following data is obtained for every slope:



(a) Slope of 10% (b) Slope of 20%

Figure 5.9: Difference error between slip vs constant

Looking at the results of the relative difference, it can be concluded that the higher the top boundary, the less influence it will have on the air flow.

If the relative difference between 0.01% and 0.1% is considered permissible, it can be seen that for the case of 10% slope, a height of 300m would be sufficient. On the other hand, for a slope of 20%, a height of 400 meters would be necessary in order to obtain the same difference.

Unless it is very close to the top, the difference between both simulations is negligible and one of them could be chosen to continue with the study of the slope. The *slip condition* is chosen since it is the one that has presented the most physical sense in the ascending slope case. Therefore, for the following simulations, $z_{top} = 300m$ of height will be selected for the case of the slope of 10% and $z_{top} = 400m$ of height for the case of 20% with the top slip condition in both cases. All in order to respect that the relative difference between both conditions is between the highlighted interval.

5.3. Flatness after ascending slope

The objective of this section is to re-incorporate the velocity profile on a flat plate in order to see its development after having encountered an ascending slope. The length is then increased by $1km$ in order to make it possible. So now the outlet of the domain will be found at $x = 1200m$.

Two slopes are chosen 10% and 20%, and with heights of $z_{top} = 300m$ and $z_{top} = 400m$ respectively, according to the conclusions of the previous case, where the relative difference between the slip and constant top conditions were between 0.01% and 0.1%.

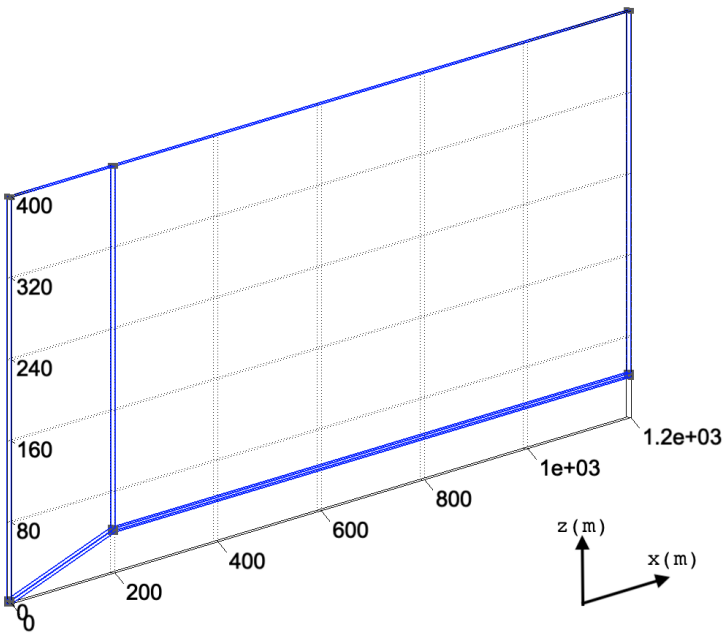


Figure 5.10: Slope of 20% ($h_{top} = 400m$)

5.3.1. Mesh regulation

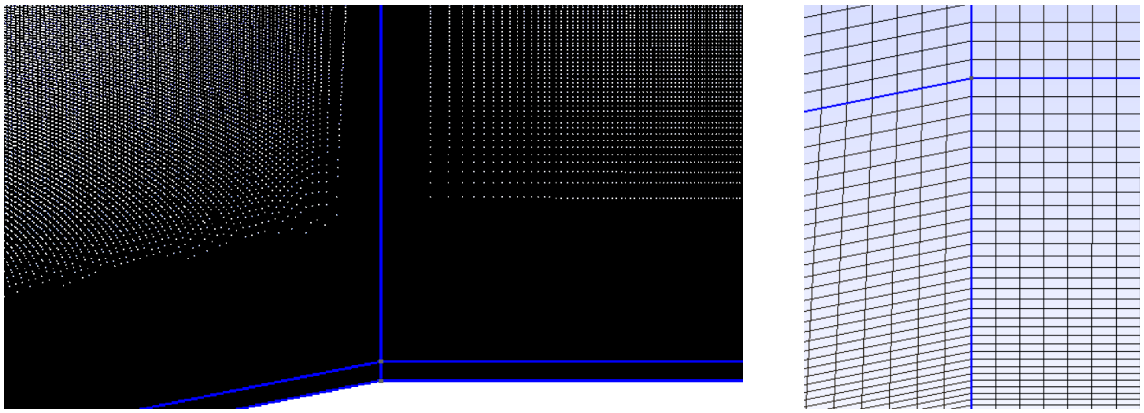
Since the computational domain now increases considerably, the number of nodes will also increase. But by showing that in the y-axis the speed does not change, the mesh will no longer be 3D but will be almost 2D. Almost 2D, since the geometry is still 3D.

As the only thing that is introduced is $1km$ more of plain, the meshing of the first $200m$ will be exactly the same as before. The reason for this decision is because of the reliability of the results. The objective is to compare the flow when reaches a slope with the flow when reaches the slope and also a plain. Therefore the meshes have to be equal in order to avoid possible changes in the results.

So the vertical direction of the geometry, the progression data, number of nodes and distance between nodes, will be respected. For the horizontal direction of the geometry, the number of nodes remains the same for the first $200m$ and a proportion is made in order to get the number of nodes in the next $1000m$ of travel:

$$N = \frac{1000m \cdot 250nodes}{200m} = 1250nodes$$

This time, the flow will undergo a change just after the slope ($x = 200m$). That's why, the progression number on the x-axis has been slightly increased, fully respecting the number of nodes as always. In this way, any turbulence or anomaly that may appear in the exchange of sections, can be controlled by the mesh:



(a) Horizontal progression (b) Zoom

Figure 5.11: Slope and flatness mesh ($x = 200m$)

This concentration of nodes will help the simulation to specify the exchange of speed and turbulence that may appear in the change of the domain. The progression corresponding to each distance, for each number of nodes is as follows:

N^o nodes / x(m)	r_h
250 / 200	1.0093
1250 / 1000	1.0020

Table 5.4: Horizontal progression data

Less progression for $1000m$ because it is an inverse proportion. The more distance, the less progression in order to guarantee the same separation of nodes at $x = 200m$.

Once the meshing has been done, the number of total elements are compiled in the following table:

Mesh Type	Total Elements
$z_{top} = 300m$ & 10% slope	1196902
$z_{top} = 400m$ & 20% slope	1296902

Table 5.5: Slope + flatness elements

5.3.2. Outcomes

Given that different velocity profiles of different heights are now being compared, what is proposed is to remove the dimension of the height in order to compare both results in real size. As simple as dividing all the points of the z-axis by the maximum value:

$$z^{'} = \frac{z}{z_{top}}$$

Where $z^{'}$ is the non-dimensional z-points, z are all the z-points and z_{top} is the maximum value of z.

Once the simulations are done, the first thing to compare is the velocity profile at $x = 200m$ between the results that considered only the slope with those that now consider the slope and the flatness. It is compared at $x = 200m$ because it is the point where the geometry stops being the same as in the previous case. It is a simple way to see the influence that the plain has now on the development of the velocity profile.

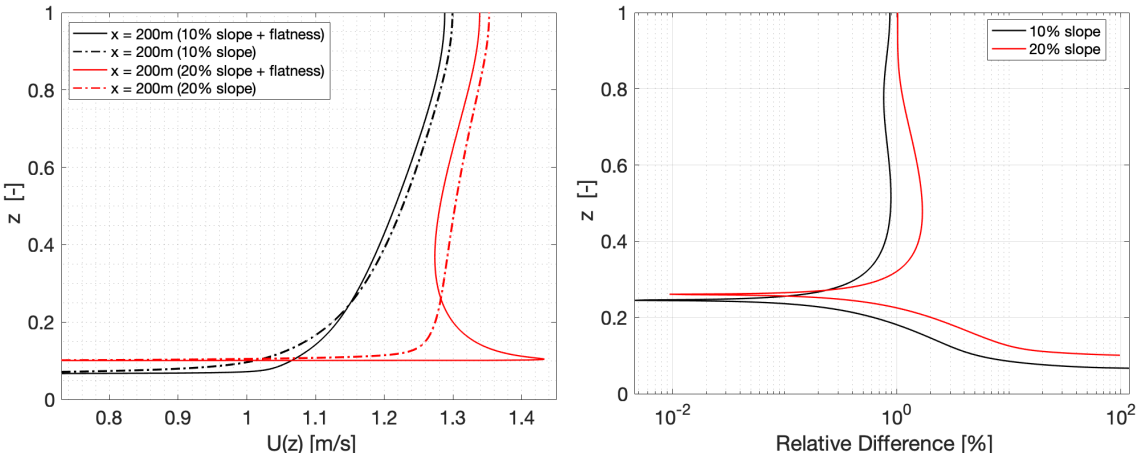


Figure 5.12: Difference between slope vs slope with flatness.

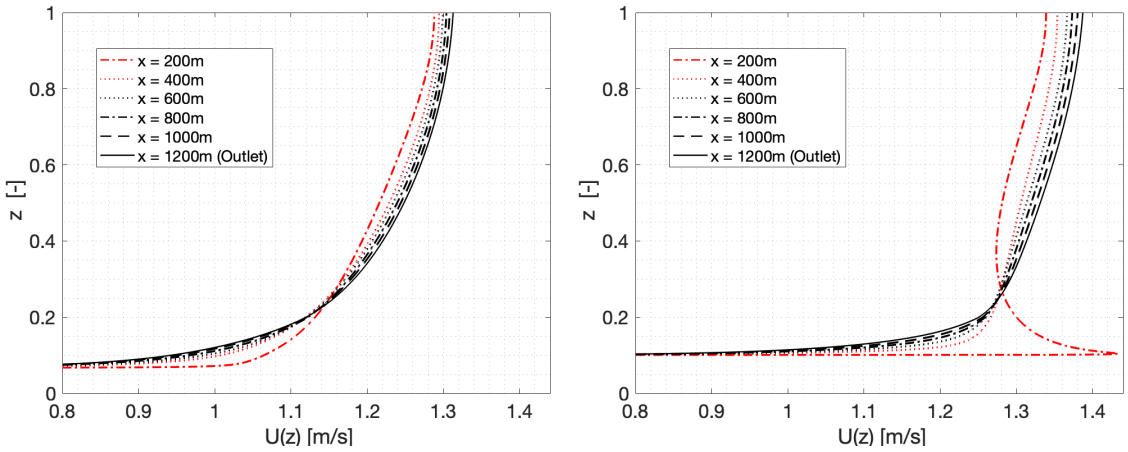
From the relative difference, large differences can be observed in the lower part of the profile. When the plain is present, the profile accelerates noticeably on both slopes. The more slope the more difference there is, since the acceleration is greater.

Since the flow is subsonic, when it passes from a convex corner to a flat plain of lower cross section, the fluid will tend to accelerate. This increase in speed in the lower part of the profile is compensated by braking it along the z-axis. The maximum speeds are detected in:

Slope $(\%)$	Height (m)	Velocity (m/s)
10	$1 \cdot 300 = 300$	1.288
20	$0.105 \cdot 400 = 42$	1.433

Table 5.6: Data at $x = 200m$.

Once past the slope, let's see the influence of the plain and how it recovers the logarithmic form:



(a) $Slope = 10\%$ $(z_{top} = 300m)$ (b) $Slope = 20\%$ $(z_{top} = 400m)$

Figure 5.13: Development of the velocity profile in a flat land after the slope.

Since the 10% slope is less pronounced, the acceleration of flow, which is obtained in the lower part just before starting the plain, is not very significant. And this makes the plain that comes after, get stabilise the velocity profile, seeking to take the form of the ABL equation.

On the other hand, in the case (B), since the slope is steeper, the velocity obtains an abrupt peak of acceleration in the lower part. And although the plain manages to stabilise the profile, it fails to take the form of the ABL equation in these 1200 meters. So it should take a lot of distance in order to get it.

In order to see the differences between the output speed and the ABL theoretical profile, the principle of conservation of the mass must be applied. A different height range than the input is found, so the theoretical profile must be adapted to these new conditions. The initial height is $z_g = 20m$ or $z_g = 40m$ and the final height is $z_{top} = 300m$ or $z_{top} = 400m$. The process to determine the ABL equation from experimental data is explained in [Annex B]. The error is as follows:

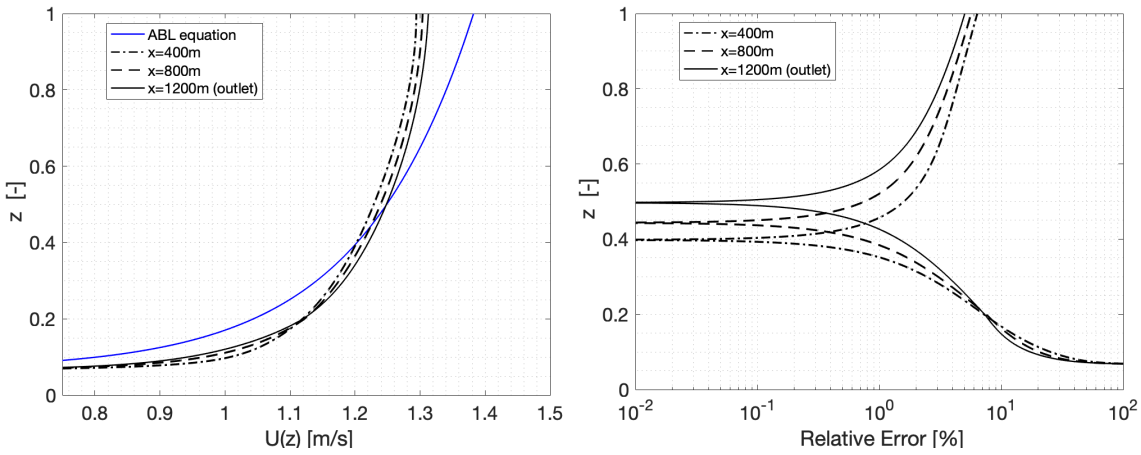


Figure 5.14: Velocity profiles after the 10% slope versus the ABL equation.

From the plot of the relative error, it can be seen that the more distance traveled, the less error it presents in the comparison with the theoretical equation. The average

error is between $\epsilon(\%) = 1$ and $\epsilon(\%) = 7$. So it would take some more travel to reach less error.

For the case of the 20% slope, the flow is greater and the ABL equation expands a little more. The comparison between the development of the profile and this one is:

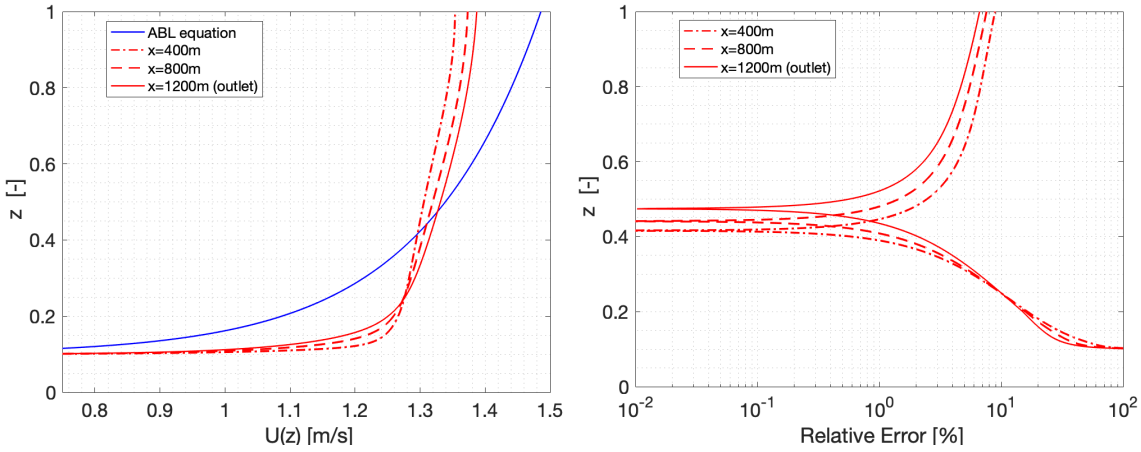


Figure 5.15: Velocity profiles after the 20% slope versus the ABL equation.

The error is more pronounced and even the development does not achieve the expected logarithmic form. The error range is between $\epsilon(\%) = 3$ and $\epsilon(\%) = 10$. It is true that the more it is traveled the closer it is got the profile, but for a distance of 1km, the profile develops quite slowly.

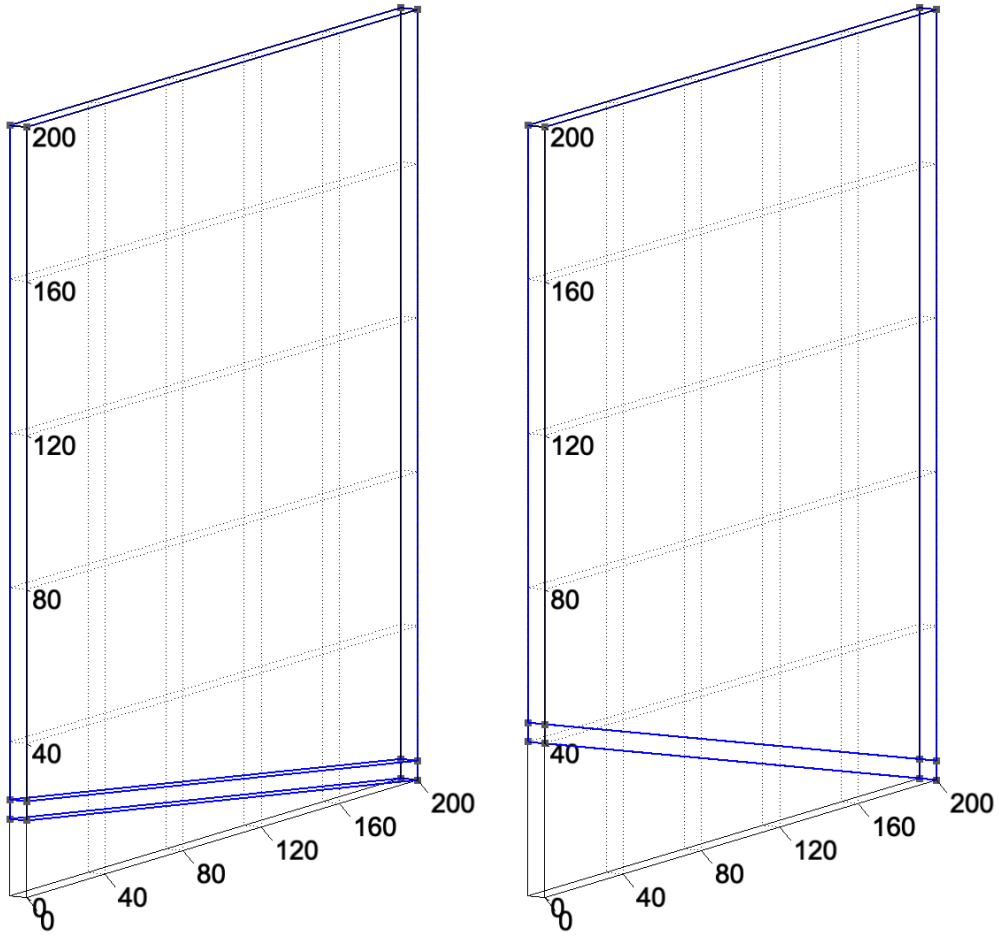
This may be due to the fact that the velocity at $x = 200m$ is quite pronounced and amorphous, so it will be more difficult for the flatness adapt the profile to the logarithmic shape.

It is now proposed to take the speed at the output of this domain and enter it into the basic geometry of always. In this way it will be possible to see if the approximation of the top cover or the 200 meters more of route are able to improve the approach.

According to the results obtained in [Annex D], it can be seen that the more flow rate, the slower the velocity profile develops. And since the profile of the output of the results in Figure [5.15] is now being chosen as an entry profile, and therefore is a profile with better appearance than at $x = 200m$, it is easier for the simulation to achieve the logarithmic profile.

5.4. Descending Slope

It is now proposed to carry out the same procedure but with descending slopes. The idea is to maintain the same geometric characteristics except the change of the slope. It is a reverse process to the previous one, but respecting the same steps to follow. The values of the slopes will continue to be 10% and 20% of the transversal length of 200m of travel.



(a) $Slope = 10\%$ (b) $Slope = 20\%$

Figure 5.16: Slopes implementation by GMSH ($z_{top} = 200m$)

5.4.1. Mesh regulation

In order to ensure that the meshing respects the same characteristics as in the ascending case, it has been maintained exactly the same values of the number of elements, vertical distances and geometric progressions. Simply, the position has been exchanged and where before corresponded to the inlet, now corresponds to the outlet. And since it has already been proven, both for the case of the horizontal plain and for the case of the ascending slope, the number of layers on the y axis has been reduced. Instead of having 5 internal layers, now what it has is simply the front layer and the back layer. So instead of multiplying by 6 the number of elements that are in

the 2D area, multiply by a factor of 2:

	Inlet and Outlet Column			Total Elements
	$h_{inlet}(m)$ / $h_{outlet}(m)$	r_{inlet} / r_{outlet}	n	$2 \cdot [(135+n) \cdot 250]$
$z_{top} = 100m$	75 / 95	1.0080 / 1.0095	200	$\approx$ 167500
$z_{top} = 200m$	175 / 195	1.0040 / 1.0045	400	$\approx$ 267500
$z_{top} = 300m$	275 / 295	1.0030 / 1.0030	600	$\approx$ 367500
$z_{top} = 400m$	375 / 395	1.0023 / 1.0024	800	$\approx$ 467500

Table 5.7: Slope 10%

For the case of 20% it is exactly the same but changing the number of the progression in the inlet.

	Inlet and Outlet Column			Total Elements
	$h_{inlet}(m)$ / $h_{outlet}(m)$	r_{inlet} / r_{outlet}	n	$2 \cdot [(135+n) \cdot 250]$
$z_{top} = 100m$	55 / 95	1.0070 / 1.0095	200	$\approx$ 167500
$z_{top} = 200m$	155 / 195	1.0038 / 1.0045	400	$\approx$ 267500
$z_{top} = 300m$	255 / 295	1.0029 / 1.0030	600	$\approx$ 367500
$z_{top} = 400m$	355 / 395	1.0021 / 1.0024	800	$\approx$ 467500

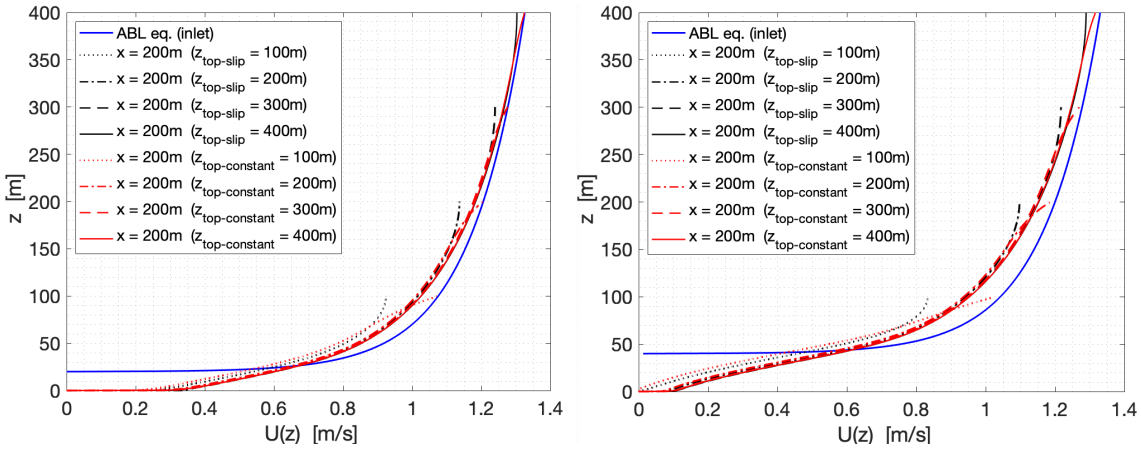
Table 5.8: Slope 20%

5.4.2. Outcomes

Once the meshes have been modified, the simulations are getting started. Notice that it is assumed that the outlet speed in the plain of the previous section, has totally converged. That is, the initial profile before going down the slope, is the theoretical velocity ABL.

Now, the velocity profile starts from a height of $z_g = 20m$ in the case of the 10%, and from a height $z_g = 40m$ in the case of the 20%. So they will be specified in the initial conditions of the velocity.

This is important because for the simulation and for the initial conditions of the top constant cover, the speed must be known. If an initial value of the wrong speed is entered, the results would not be output correctly. Using then the ABL equation (1.12) by imposing that for 10% is $z_g = 20m$ and for 20% is $z_g = 40m$, it is obtained:

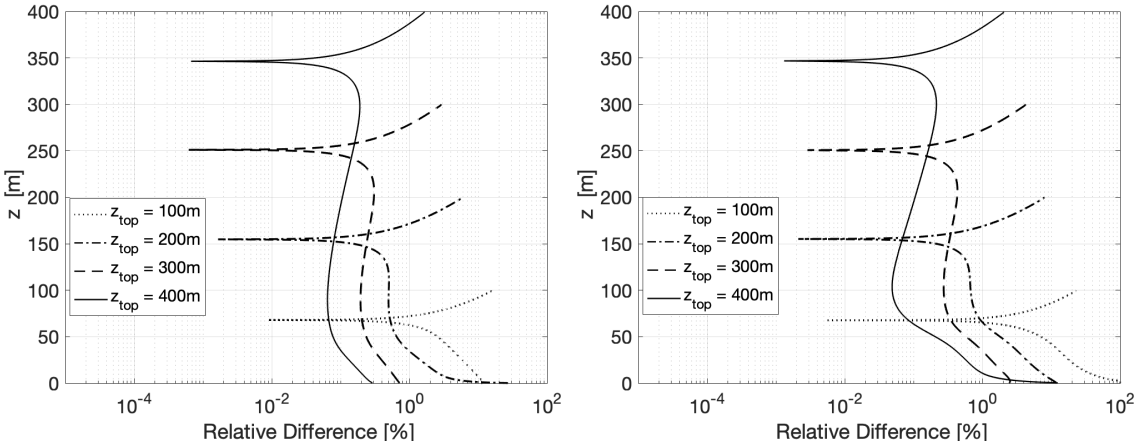


(a) Slope of 10% (b) Slope of 20%

Figure 5.17: Outlets of different slopes with different top boundary conditions.

In this case is the other way around, the more slope, the more the velocity profile is slowed and the more it separates from the initial profile. In the lower part it can be seen how the influence of the roughness intensifies, ”raising” the profile with a tendency to be reduced.

In order to observe the differences between the slip condition and the constant condition, the relative difference is computed like in the previous case (5.3):



(a) Slope of 10% (b) Slope of 20%

Figure 5.18: Difference error between slip vs constant

From the relative difference can be seen that for the case of the descending slope, the difference is greater than for the case of the rising slope, whether it is 10% or 20%. It is also closer to the difference of 0.1% than for 0.01%.

The lower part of the velocity profile there is more inequality. This may be due to the sum of the decelerations that the profile is suffering, both on the part of the roughness, and that of being in a more extended domain. Because the more slope, the more difference is observed in that region.

Finally, both slopes (10% & 20%) will need a height of $z_{top} = 400m$ in order to provide that the relative difference remains between the selected interval of 0.1% and 0.01%.

5.5. Flatness after descending slope

Once the height for both slopes has been found, the length of the x axis is increased by 1km in order to see the development of the velocity profile, after having braked due to the roughness and the expansion of the computational domain. The geometry is then as follows:

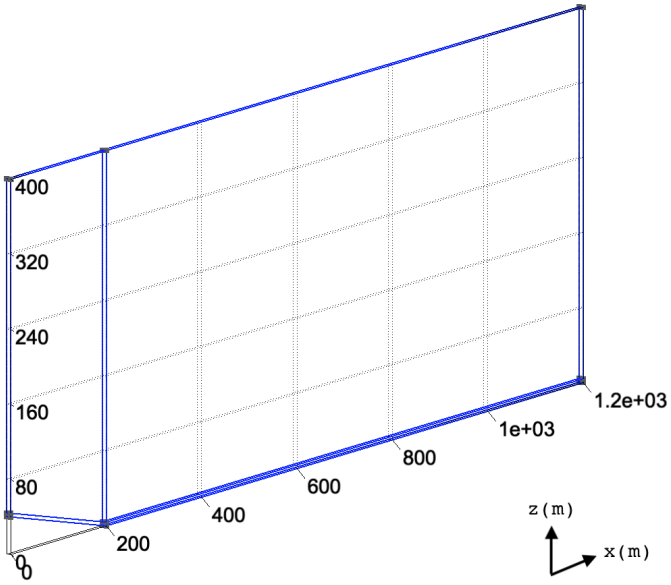


Figure 5.19: Descending slope and flatness

5.5.1. Mesh regulation

The same proportion of data for the number of nodes, geometric progressions and distances are considered for the new mesh in question. There are hardly any changes, as it maintains the same philosophy as always for both of them.

The only difference is that in the case of 10% slope, the mesh has increased due to having selected the height geometry of $z_{top} = 400m$. So now, there is the same amount of elements for both slopes:

Mesh Type	Total Elements
$z_{top} = 400m$ & 10% slope	1296902
$z_{top} = 400m$ & 20% slope	1296902

Table 5.9: Slope + flatness elements

The only thing that will vary from the two meshes will be the geometric progression and the vertical distance of entry.

5.5.2. Outcomes

As in this case, it is working with the same top heights.Therefore, all simulations will be shown with height in meters.

Once the simulations have been made, what will be compared now is the coincidence point at $x = 200m$, in order to see the influence that the plain has on the decelerated profile. The results are shown below:

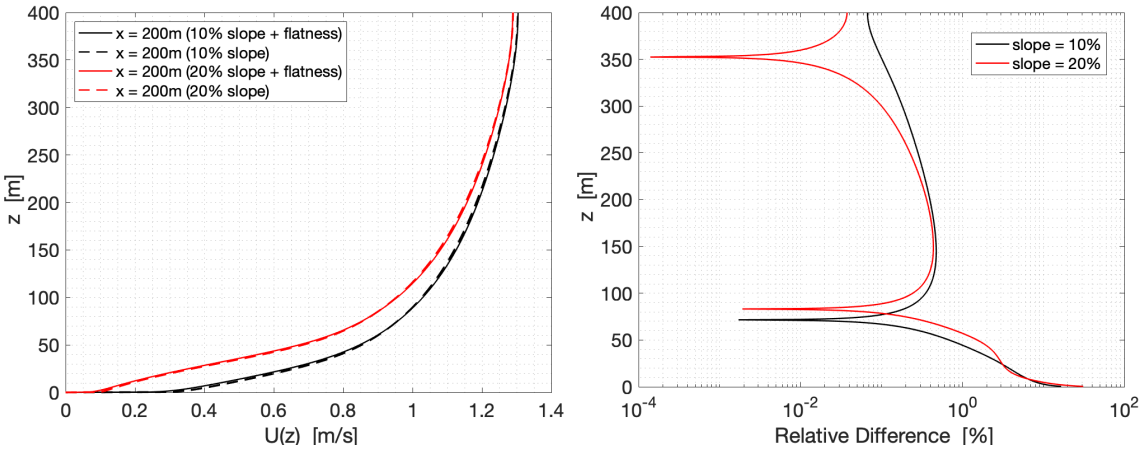


Figure 5.20: Difference between slope and slope with flatness.

This time, observing the results in detail, it can be seen that appears the inverse effect that appeared in the case of the ascending slope. When the flatness is added, the flow tends to accelerate, slightly, through the upper part of the profile. On the other hand, in the lower part it can be seen that the speed tends to be reduced.

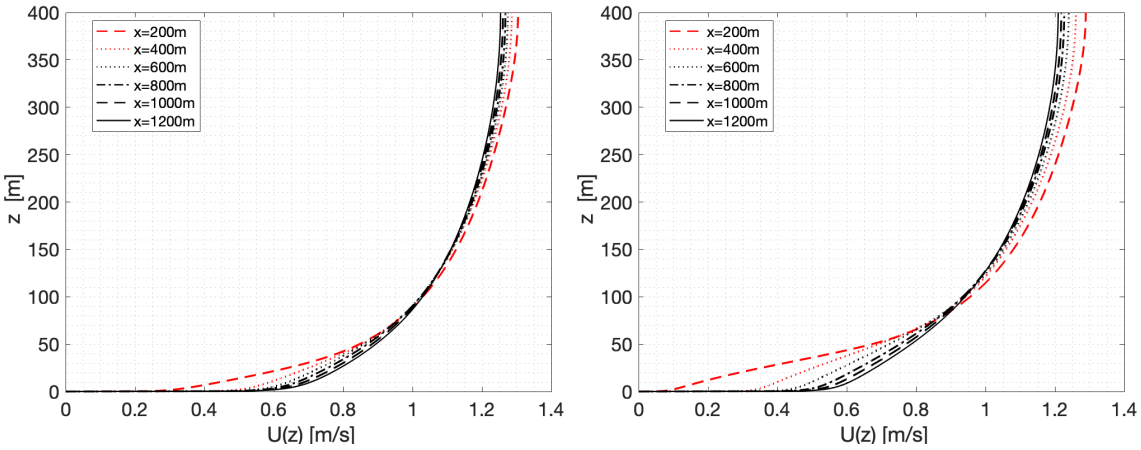
In the plot of the relative difference, a similarity between both slopes between $z = 125m$ and $z = 200m$ is shown. It could be said that in that range of heights, the influence of roughness and flatness on the flow is practically the same for both the 10% and the 20%.

For the first 75 meters of height, the slope of 10% presents less percentage of difference, and between the remaining $z = 100m$ and $z = 400m$ of height, the 20% is smaller.

To sum up, the highest relative difference is found in the bottom of the profiles. Therefore the plain, continues to have more influence on the ground than in the rest of heights.

Now it would be necessary to see the development of the profile through the plain. In the case of the ascending slope, the plain tended to slow down the lower part and accelerate it through the upper part, getting it to adopt the logarithmic form little by little, although it did not reach it completely.

Let's see the results of the development for this case:



(a) $Slope = 10\%$ $(z_{top} = 400m)$ (b) $Slope = 20\%$ $(z_{top} = 400m)$

Figure 5.21: Development of the velocity profile in a flat land after the slope.

As it advances on the plain, the speed is regulated along the z-axis, gradually adopting the logarithmic form. It increases at the bottom and decreases at the top. Even though the speed has increased for the case of the slope of 20%, this is still lower than the 10%, for the same route and in the same time of transfer. So the reduction is globalised along the z-axis.

Visually, the logarithmic form seems to reach it, but to ensure this affirmation, part of the development is compared with the theoretical ABL equation.

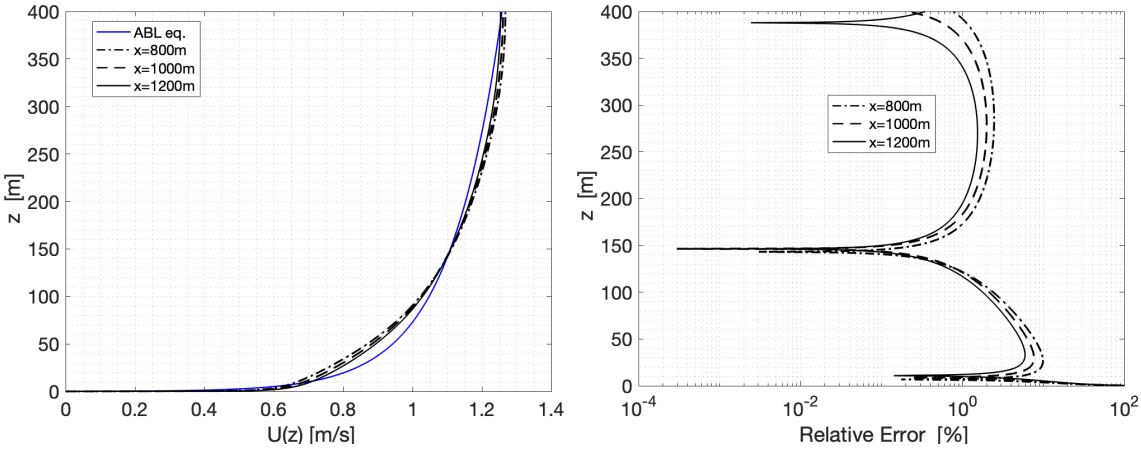


Figure 5.22: Velocity profiles after the 10% slope versus the ABL equation.

From the plot of the errors, it can be seen that it is in a range between $\epsilon(\%) = 1$ and $\epsilon(\%) = 7$ error. They are ranges of errors similar to those obtained in the upward slope of 10%. That means, that both for ascending slope and descending slope, the profile gets close enough to the theoretical profile.

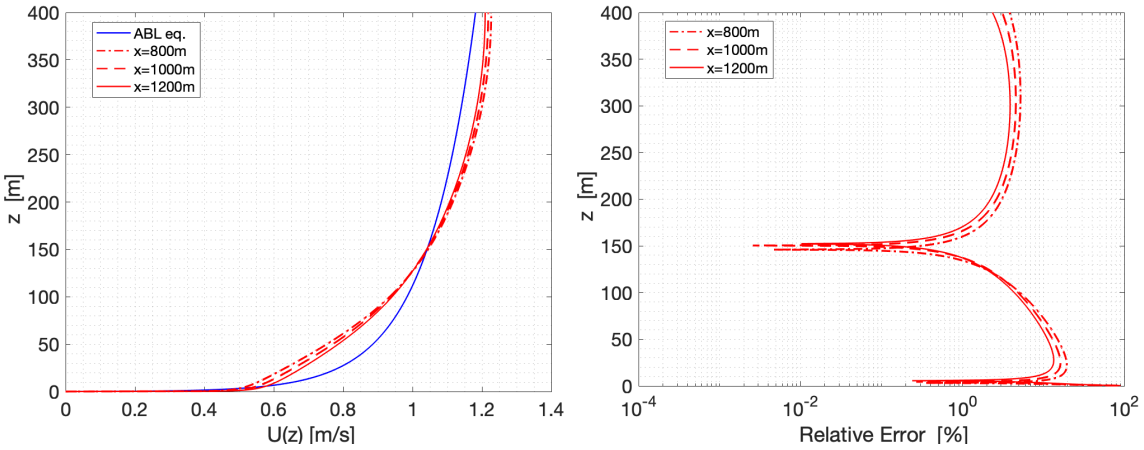


Figure 5.23: Velocity profiles after the 20% slope versus the ABL equation.

On the other hand, in the case of 20%, it can be seen that the error is more pronounced, more close to $\epsilon(\%) = 7$ error , so that more distance would be needed provided the profile got closer to the logarithmic form ABL.

To sum up, it can be said that the more slope there is, the harder it is to reach the theoretical logarithmic profile.

5.6. Discussion and final remarks

5.6.1. OpenFOAM

This software has no visible interface, since it works with folders and files programmed in C++. Therefore, the user will spend much time trying to understand the functionality of OpenFOAM.

It is quite dependent, that is, it needs the help of other programs in order to carry out the global CFD process. The pre-processing can be done within the program but it is very difficult to understand, so it is easier to use an auxiliary program for it (GMSH). For the post-processing it will be the same, it is needed another auxiliary program (Paraview) in order to see the results obtained.

OpenFOAM offers different solvers to be used, that is, in order to carry out the simulation process, a specific solver must be selected according to the characteristics of the fluid. So the user must adapt to what OpenFOAM offers.

It is a tool quite suitable, but with some drawbacks for the person who is new to the subject. If the learning is totally autonomous, it will require a great investment of time to study the operation and procedures that OpenFOAM has. If the user has a teacher or a referent of the subject, the user will be able to elaborate more quickly all the procedures required.

5.6.2. Preprocessing

The factor of the boundary conditions is a great importance in the development of the velocity profile. When a uniform inlet is introduced, the simulation is not able to obtain a logarithmic profile. On the other hand, when the theoretical profile is introduced, the simulation hardly changes in the results.

A study about the ”Assessment of Inflow Boundary Conditions for RANS Simulations of Neutral ABL and Wind Turbine Wake Flow” [27], gets the following conclusion:

”*The turbulence decay problems do exist as expected when employing the uniform inflow conditions for ABL simulations with RANS method. Besides, the predictions under un-uniform conditions outperform the results under uniform conditions, with respect to both wind speed and turbulence quantities profiles.*”

Therefore, the affirmation agrees with the work.

5.6.3. Work tools

The use of a MAC computer is a disadvantage, due to the incompatibility problems that arise between the operating system with the programs that the work needs. In order to do an effective job, the Linux operating system is required.

The computers used have little power, so the geometries created in 3D and the meshes that are generated within them, can be an inconvenience in the calculation delay. Therefore, in order to achieve efficiency and fluency in the performance of

work, it would be best to acquire a computer with a Linux operating system with a large processing memory. In this way, the user will be able to install all the necessary software with ease. And the simulations will run faster.

The calculation time of the simulations of this project has been very large, from around 1 week for the case of 3D geometries and between 3 to 5 days for each 2D simulation. That is why a powerful computer is necessary.

5.6.4. Results

The influence of roughness has an important role in the development of the velocity profile, as well as in its logarithmic expansion. The more roughness, the more friction and therefore the less speed there will be. The use of wall functions to represent the roughness is a problem in the lower part of the computational domain for the fitting of the profile. An investigation on "Appropriate boundary conditions for computational wind engineering models revisited" by S. E. Norris a, P. J. Richards made in New Zealand at University of Auckland, wise in Mechanical Engineering [25], they concluded the following:

"*Using the $k-\epsilon$ turbulence model exhibit a characteristic maxima in the value of k near the ground, rather than the constant value across the layer predicted by the analytic theory. This is shown to be due to the discretisation of the turbulence production.*"

According to the results obtained, it can be said that it is true that the peak of error is due to turbulence, as it has been seen in the section [2.2.1.]. But really the problem is because of the combination between the wall functions and the mesh. where the CFD program will not be able to accurately represent the velocity profile.

Another important aspect to take into account is the condition that must be imposed on the top boundary of the computational domain. A research on "CFD simulation of the atmospheric boundary layer: wall function problem" [26], says that:

"*Specific attention is needed for the boundary condition at the top of the domain. Along the length of this top boundary, the values from the inlet profiles of U, k and ϵ at this height are imposed. The application of this particular type of top boundary condition is important because other top boundary conditions (symmetry, slip wall, etc) can themselves cause stream-wise gradients, in addition to those caused by the wall functions.*"

According to the results that expose the differences between the constant top and the slip top condition, it can be said that it is true that the top cover must be studied with caution. In the case of the flatness geometry, it can be seen that the constant top, adjust more the velocity profile with respect to the sliding top. But the top slip condition is inevitable, since for other more sophisticated geometries, no CFD user is able to know what values of turbulence and speed correspond to the top boundary. That is, in the case of having slopes and unevenness in the terrain, the appropriate value that would correspond, it comes through the progress of the simulation. Imposing a constant velocity value on the top, would corrupt the flow of air for geometries that need other velocities. In the figure [5.5], it is a clear example that shows that the constant condition in the top is not effective for the geometric change. The sliding condition is more physically viable and probable.

By increasing the height of the geometries, the conditions of the top cover weaken. The advantage of this strategy is that the user should not worry about the condition that must be imposed. Since the results are practically the same and researchers can better focus on the main objective that concerns them.

The greater the slope, the greater the acceleration of the flow. The built-in plain just after the ascending slope is responsible for stabilising the velocity profile. But once the flow has traveled $1km$ away, the profile can not easily return to the theoretical logarithmic form. On the other hand, in the case of the descending slope, it is possible to adapt the logarithmic form much better.

According to the extra simulation performed in [Annex D], speed tends to get closer to the logarithmic form. So 200 meters more travel with less amount of flow, since the height of the geometry is 50m in this check, it is easier to reach.

Based on these results, it can be said that this difficulty to stabilise the flow (obtain the logarithmic form), may be due to the fact that the plain starts from a logarithmic profile already accelerated and therefore would need more distance to achieve it. Because in the case of the descending slope, the flow accelerates more easily in 1 km, getting closer to the theoretical profile. Therefore, at higher speed, it is more difficult to brake the flow and therefore more distance would be needed on the transverse axis. The more amorphous the velocity profile at the beginning of the plain, the more difficult it will be to recover the theoretical logarithmic form. And the more cross section, the more difficult to develop the logarithm.

CONCLUSIONS

In this work, the influence that the orography of the terrain had on the propagation of the wind is studied. For this, the GMSH and OpenFOAM programs have been used: the first, in order to be able to generate meshes within any geometric domain; the second, in order to be able to calculate the velocity in each node of the mesh. The equations that are applied in each node are simplified using the SimpleFoam solver, since it takes into account the physical characteristics of the chosen fluid: Incompressible, steady-state, turbulent, single phase and isothermal. Once the best boundary conditions are selected, we proceed to study the influence of the roughness as well as the different geometries used: plain, ascending slope, ascending slope with plain, descending slope and descending slope with plain. The following conclusions about the work are obtained:

- The use of ”wall functions” is useful for the representation of the roughness. But its misuse can cause errors in the formation of the velocity profile. The meshing and the ”wall functions” are basic and necessary concepts in order to obtain reliable simulations. A fine mesh ensures precision in the results but does not avoid the Buffer Layer in the lower part of the domain. Increasing the friction velocity causes the values of y^+ to increase, so it moves away faster from the Buffer Layer.
- The higher the roughness, the greater the friction and therefore the slower velocity. And the more roughness, the greater the value of the turbulence.
- An ABL inlet will always be better than a uniform inlet. And a top slip condition will allow the flow to progress with more degree of freedom than a constant condition. The higher the top boundary, the less influence it will have on the rest of the fluid.
- A constant upward slope, accelerates the flow throughout its domain. On the other hand, an ascending slope together with a plain, accelerates the flow through the lower part of the profile and reduces its speed in the rest of the heights.The more percentage of slope there is, the more this conclusion will be intensified.
- On the other hand, constant downward slope, decelerate the flow throughout its domain. An descending slope together with a plain, decelerates the flow through the lower part of the profile and increases its speed in the rest of the heights. The more percentage of slope there is, the more this conclusion will be intensified.
- The resulting velocity profile just after the slope, has difficulties recovering the theoretical logarithmic form along the path at x-axis. The more percentage of slope there is, the more this conclusion will be intensified.

This work provides essential concepts for the correct elaboration of a simulation. And to demonstrate that the results are reliable, a direct study could be made without fragmenting the different geometries by sections. Since the simulations took a long time, this last check could not be made due to lack of time. So the idea is proposed in case someone else could do it.

BIBLIOGRAPHY

[1] ESA. "Space in Videos". (*https://www.esa.int/spaceinvideos/Videos*) xi, 3

[2] ESA's wind mission " Aeolus ". https://m.esa.int/ESA. 3

[3] Wei Tong. "Wind Power Generation and Wind Turbine Design". (*Technology & Engineering*) (2010) 4

[4] Project leader: János VAD DSc " Project: Establishment of an Atmospheric Flow Laboratory" *Budapest University of Technology and Economics.* (Submission of the project proposal: January 4th, 2017) xi, 4, 5, 45

[5] DTU Wind Energy. "Different Roughness". (*https://www.youtube.com/channel/UCcqkYU4XUkU$_G$nGHdjhIy8A*) xi, 5

[6] Zhang, Xiaodong. "CFD simulation of neutral ABL flows". *Danmarks Tekniske Universite*. (2009) xiii, 6, 12

[7] National Aeronautics and Space Administration. "U. S. Standard Atmosphere" *United States Air Force & United States Weather Bureau* (1962) 7

[8] John D. Anderson, Jr. " Fundamentals of Aerodynamics " (Fifth Edition) (2011) (Previous editions in 2007, 2001, 1991 and 1984) xi, 7, 8, 9

[9] Popular Mechanics. "Turbulence over mountains". (*https://www.popularmechanics.com/flight/a5354/4327148/*) xi, 10

[10] Modelling Turbulent FlowsPopular Mechanics. "Introduction FLUENT Training". (*www.fluentusers.com*) 10

[11] Launder, B.E; Spalding, D.B (March 1974). "The numerical computation of turbulent flows". *Computer Methods in Applied Mechanics and Engineering.* (269-289) 10

[12] Versteeg, Henk Kaarle; Malalasekera, Weeratunge (2007). *An introduction to Computational Fluid Dynamics: The Finite Volume Method.* Pearson Education. 10

[13] P.J. Richards, R. Hoxey,. "Appropriate boundary conditions for computational wind engineering models using the $k-\epsilon$ turbulence model". *J. Wind Eng. Ind. Aerodyn. 46 & 47.* 145-153 (1993) 11

[14] Christopher J. Greenshields and The OpenFOAM Foundation. "User Guide version 6".http://openfoam.org. (2018) 15, 16

[15] H. Tennekes & J.L. Lumley. "A First Course in Turbulence". Cambridge, Massachusetts, and London, England. (1972) 16

[16] Tsan-Hsing Shih, Louis A. Povinelli, Nan-Suey Liu, Mark G. Potapczuk and J. L. Lumley. "A Generalized Wall Function" *NASA Center for Aerospace Information & National Technical Information Service* (April 21, 1999) 16

[17] Cebeci, T. and Smith, A.M.O. (1974). "Analysis of turbulent boundary layers" *Academic Press, New York* 18

[18] "Set Partitions: String Numbers". *Digital Library of Mathematical Functions.* Retrieved 24 May 2018. 25

[19] Iehisa NEZU, Akihiro TOMINAGA (2000): ”Suirigaku”, Asakura Shoten, pp.130-133 xi, 19

[20] ”CFD Support”, Boundary Conditions. *https://www.cfdsupport.com* xi, 21

[21] Philippe R.Spalart and Christopher L. Rumsey. “Effective Inflow Conditions for Turbulence Models in Aerodynamic Calculations”. *American Institute of Aeronautics and Astronautics*. pagina2–pagina10. (2007) 40

[22] Scientific & Academic Publiching. ”Ascending slope”. $(http://www.sapub.org/journal/index.aspx)$

[23] OpenFOAM User Guide. ”SimpleFoam definition” $(http://www.openfoam.com)$ 35

[24] Mathematical Modeling and Experimental Verification of Fluid Flow through Deformable Rough Rock Joints. $www.researchgate.net/publication/273405250$ (July 2015) xi, 36

[25] S. E. Norris , P. J. Richards. ”Appropriate boundary conditions for computational wind engineering models revisited” *University of Auckland, New Zealand*. (May 2010) 62

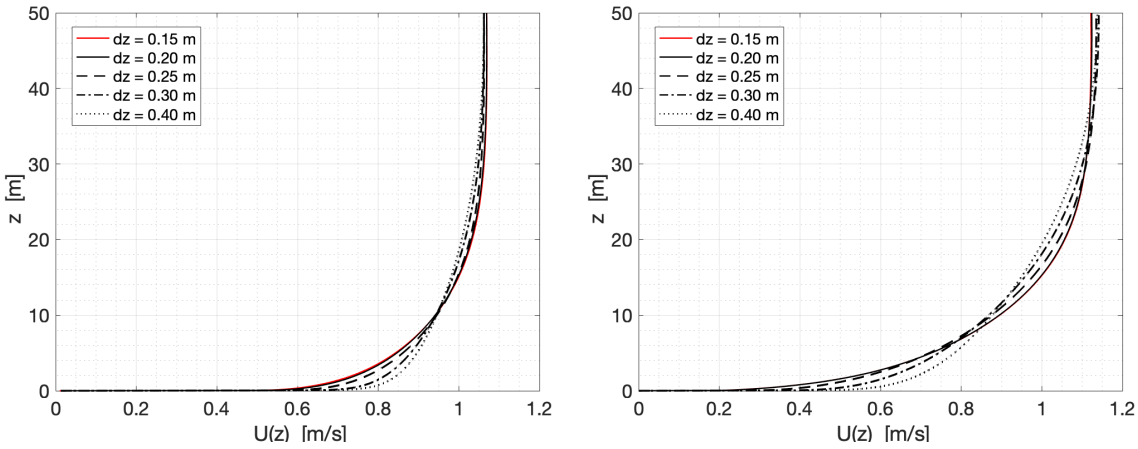
[26] Bert Blocken , Ted Stathopoulos , Jan Carmeliet ”CFD simulation of the atmospheric boundary layer: wall function problems” *Technische Universiteit Eindhoven*. (2007) 62

[27] Linlin Tian, Chunling Zhu, Weijun Zhu and Ning Zhao, ” Assessment of Inflow Boundary Conditions for RANS Simulations of Neutral ABL and Wind Turbine Wake Flow” *Nanjing University of Aeronautics and Astronautics* and *School of hydraulic energy and power engineering*. (28 July 2017) xi, 23, 61

APPENDICES

APPENDIX A. ADDITIONAL MESH CONVERGENCE RESULTS

The convergence of the mesh has also been done for different roughnesses.



(a) Outlets at $x = 200m$ with $z_0 = 1cm$

(b) Outlets at $x = 200m$ with $z_0 = 100cm$

Figure A.1: Results depending on the characteristics of the mesh.

APPENDIX B. ABL PROFILE FROM EXPERIMENTAL DATA

Consider the control volume with its inlet (green) and outlet (red) flow:

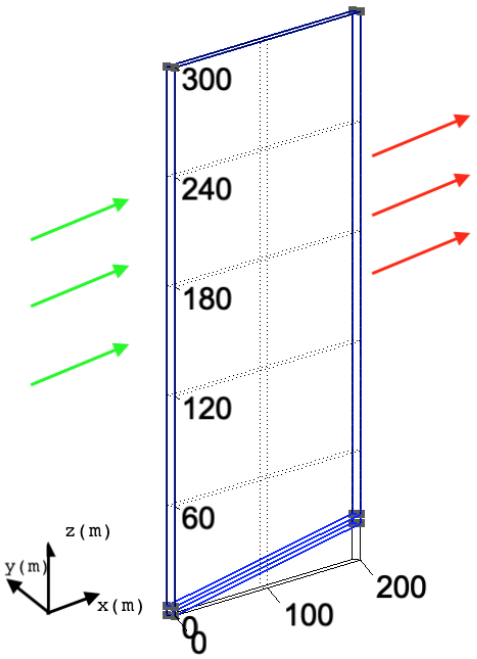


Figure B.1: Inlet and Outlet of the flow

The aim is to obtain a theoretical velocity profile that respects the same kinetic energy as that presented by another velocity profile at any x-point, for example at the outlet of the domain. So the mass conservation must comply:

$$\iint_{S1} \rho_1 U_1 dS_1 = \iint_{S2} \rho_2 U_2 dS_2 \tag{B.1}$$

$$dS_1 = dy_1 \cdot dz_1$$

$$dS_2 = dy_2 \cdot dz_2$$

A slope of 10% is found with the top at $z_{top} = 300m$ in height. As the air is incompressible and the change of the velocity does not depend on the depth of the y-axis, it can be neglected and it follows the next equation:

$$\cancel{\int_{y_0}^{y_f} \rho_1 dy_1} \cdot \int_{z_g}^{z_{top}} U_1 \cdot dz_1 = \cancel{\int_{y_0}^{y_f} \rho_2 dy_2} \cdot \int_{z_g}^{z_{top}} U_2 \cdot dz_2 \tag{B.2}$$

$$\int_{z_g}^{z_{top}} U(z) dz = \int_{z_g}^{z_{top}} U_{exp}(z) \cdot dz \tag{B.3}$$

Where $U_{exp}(z)$ is the experimental data collected in a vector, $U(z)$ the theoretical velocity that is unknown and z_g is the minimum "z" coordinate desired in any point "x".

From Matlab, velocity profile data in $x = 200m$ ($z_g = 20$) is collected in two vectors:

- $Z = [20, 20.05, 20.1, 20.15, ..., 300]$
- $UZ = [0, 0.1, 0.2, 0.3, ..., 1.21]$

In order to calculate the integral of the experimental data, the "trapz" Matlab function is used. "trapz (X, Y)" integrates Y with respect to the coordinates or the scalar spacing specified by X. So in this case it would be: "$Q = trapz(Z, UZ)$". Replacing $U(z)$ by (1.12) equation, the integral now is:

$$\frac{U^\star}{\kappa} \underbrace{\int_{z_g}^{z_{top}} ln(\frac{z - z_g + z_0}{z_0}) dz}_{\Upsilon} = \int_{z_g}^{z_{top}} U_{exp}(z) \cdot dz = Q \tag{B.4}$$

$$\frac{U^\star}{\kappa} = \frac{Q}{\Upsilon} \tag{B.5}$$

Knowing that the friction velocity (1.15) gets the velocity reference U_{ref} for every Z_{ref}, it can be concluded, that the theoretical velocity profile with the same kinetic energy of the experimental velocity profile, will be:

$$\frac{U_{ref}}{ln(\frac{Z_{ref} + z_0 - z_g}{z_0})} = \frac{Q}{\Upsilon} \tag{B.6}$$

$$\boxed{U_{ref} = \frac{Q}{\Upsilon} ln(\frac{Z_{ref} - z_g + z_0}{z_0})}$$

Finally it has been possible to obtain an expression of the theoretical model from the experimental data at any point "x".

APPENDIX C. Y AXIS RESULTS

C.1. Flat case

Starting from an intermediate point of the longitudinal axis $x = 100m$, the obtained results of the axis in depth "y" are extracted. The distance of $y = 5m$ has been divided into 4 parts, so the results obtained are for each $1.25m$.

The resulting velocity profile in $y = 0m$ is taken as a reference and a relative difference is made between this result and those obtained in different distances from the "y" axis. The following is achieved:

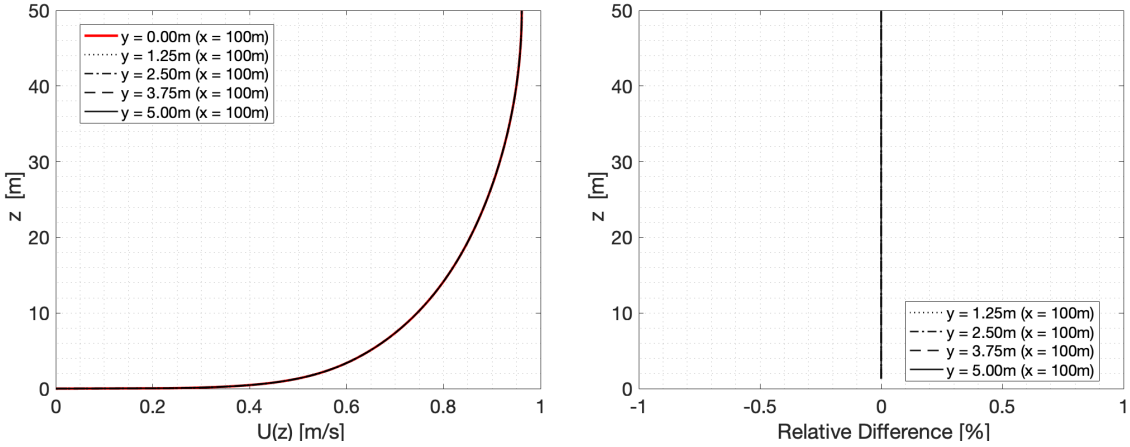


Figure C.1: y axis results (flatness)

It can be seen that the relative difference is 0% in all cases so it can be concluded that there are no changes in the speed along the axis in depth "y", and if there are modifications, they are highly negligible.

C.2. Slope case

For the case of the 10% slope with a height of $z_{top} = 300m$, it is exactly the same procedure that has been done for the case of the flat plate. It is got the following:

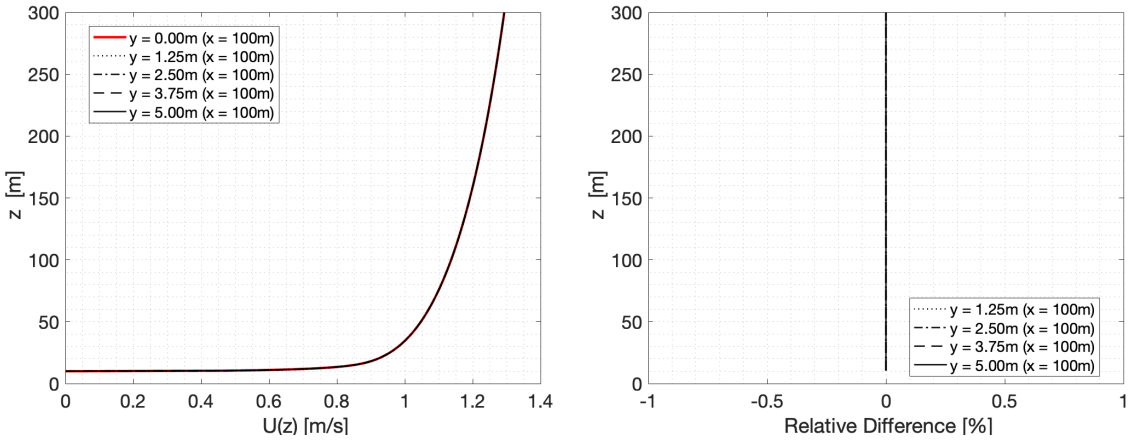


Figure C.2: y axis results (slope)

Looking at the graph of the relative difference, it can be concluded that whether or not there is slope, the velocity of the fluid is not modified along the "y" axis. Therefore it will not be necessary to continue to mesh the geometry in this direction. That way you can save calculation time in the simulations.

APPENDIX D. OUTLET PROFILE AS INPUT INTO A FLAT PLATE

It is proposed to take the outlet velocity profile of the [5.15] result, and inputting it into the basic geometry of $200m$ long and $50m$ high. It is done in order to see if the *top cover* really had an influence on the development of the velocity profile (for the case of the plain) or if it really needed more *length* to travel after having the slope.

Given that the slope of 20% is the most underdeveloped, it has been considered to make this test only for this slope. It is obtained:

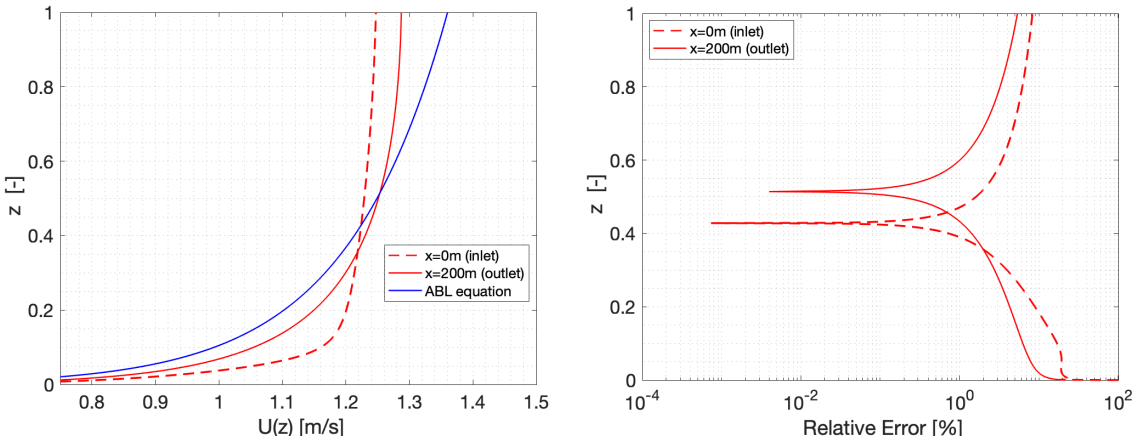


Figure D.1: Velocity profile development of the exit of the 5.15 ($z_{top} = 50m$)

It can be seen that the output in this case, does get remarkably close to the theoretical profile. Therefore, when cutting the cross section and lengthening $200m$ more of the path, the flow tends to recover the logarithmic form more easily. Less error is getting at the outlet of the domain.

For more conclusions, see section [5.3.2.].